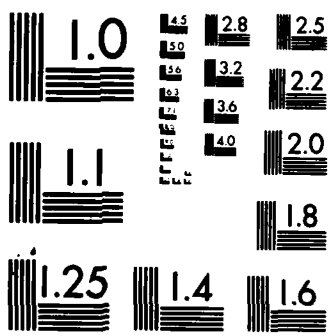
AD-A140 609 DEFENSE SWITCHED NETWORK TECHNOLOGY AND EXPERIMENTS PROGRAM(U) MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB C J WEINSTEIN 30 SEP 83 ESD-TR-83-228 1/1

UNCLASSIFIED F19628-80-C-0002 F/G 17/2 NL

END
6-84




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ESD-TR-83-228

(12)

Annual Report

# Defense Switched Network Technology and Experiments Program

AD-A140 609

DTIC
ELECTE
MAY 1 1984
S D
A

30 September 1983

## Lincoln Laboratory

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

*LEXINGTON, MASSACHUSETTS*

Prepared for the Defense Communications Agency under
Electronic Systems Division Contract F19628-80-C-0002.

Approved for public release; distribution unlimited.

DTIC FILE COPY

84 05 01 007

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, for the Defense Communications Engineering Center of the Defense Communications Agency under Air Force Contract F19628-80-C-0002.

This report may be reproduced to satisfy needs of U.S. Government agencies.

The views and conclusions contained in this document are those of the contractor and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the United States Government.

The Public Affairs Office has reviewed this report, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Thomas J. Alpert

Thomas J. Alpert, Major, USAF
Chief, ESD Lincoln Laboratory Project Office

Non-Lincoln Recipients

**PLEASE DO NOT RETURN**

Permission is given to destroy this document when it is no longer needed.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LINCOLN LABORATORY

# DEFENSE SWITCHED NETWORK TECHNOLOGY AND EXPERIMENTS PROGRAM

ANNUAL REPORT
TO THE
DEFENSE COMMUNICATIONS AGENCY

1 OCTOBER 1982 — 30 SEPTEMBER 1983

ISSUED 29 FEBRUARY 1984

DTIC
ELECTE
MAY 1 1984
S D
A

Approved for public release; distribution unlimited.

LEXINGTON MASSACHUSETTS

# ABSTRACT

This report documents work performed during FY 83 on the DCA-sponsored Defense Switched Network Technology and Experiments Program. The areas of work reported are: (1) development of routing algorithms for application in the Defense Switched Network (DSN); (2) instrumentation and integration of the Experimental Integrated Switched Network (EISN) test facility; (3) development and test of data communication techniques using DoD-standard data protocols in an integrated voice/data network; and (4) EISN system coordination and experiment planning.

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
Distribution/
Availability Codes
Avail and/or
Special
A-1
DTIC COPY INSPECTED 2

# TABLE OF CONTENTS

## LIST OF ILLUSTRATIONS

# DEFENSE SWITCHED NETWORK TECHNOLOGY AND EXPERIMENTS PROGRAM

## 1. INTRODUCTION AND SUMMARY

This report documents work performed during FY 1983 on the DCA-sponsored Defense Switched Network Technology and Experiments Program. The areas of work reported are: (1) development of routing algorithms for application in the Defense Switched Network (DSN); (2) instrumentation and integration of the Experimental Integrated Switched Network (EISN) test facility; (3) development of internetwork data communication techniques for integrated voice/data systems; and (4) EISN system coordination and experiment planning.

Routing algorithm efforts during FY 83, described in Section 2, have focused on the development of an extensive call-by-call network simulator and the application of this simulator to evaluate dynamic routing algorithm performance including the effects of a variety of Multi-Level Precedence (MLP) techniques. Performance advantages of mixed-media routing and adaptive mixed-media routing procedures after network damage had been demonstrated during FY 82 using a modified steady-state network analysis program. The call-by-call simulator now includes dynamic simulations of these routing algorithms, and also includes preemption and flooding algorithms that cannot be tested using steady-state analysis. A key result of the simulation studies shows that blocking for high-precedence users can be decreased significantly without preemption by using a technique such as Precedence Blocked Flooding (which uses flooding for only those precedence calls which are blocked with mixed-media routing). Results also indicate that the Common-Channel Signaling (CCS) bandwidth required to support precedence flooding techniques is not excessive.

During FY 83, Lincoln has also continued its major role in the development and integration of experimental subsystems for EISN (see Section 3). Packet/Circuit Interface (PCI) and Telephone Office Emulator (TOE) equipment has been installed and integrated at RADC, Ft. Monmouth, and Ft. Huachuca, so that all five EISN sites (previous installations were at DCEC and Lincoln) are equipped with this facility. Also, during FY 83 the PCI capability was augmented to support precedence and preemption. Finally, PCI voice and signaling interfaces have been designed to allow integration of the PCIs with the digital switches and Routing/Control Processors (RCPs) of the advanced EISN facility.

The FY 82 Annual Report described the preliminary design of an advanced EISN facility including commercial switches and outboard RCPs to allow EISN experimentation with the new routing and MLP algorithms. Detailed design and development of the RCP/switch facility has been a major FY 83 effort, as described in Section 3.3. Two RCP/switch systems are currently operational at Lincoln Laboratory, including custom interfaces to monitor

and control the switches, and software to control local calls. The basic RCP capability to control local calls and to accommodate precedence features has been demonstrated. Implementation of new routing algorithms in the RCP has been initiated and wlll be a major activity for FY 84.

The goal of the EISN data communication experiments (Section 4) is to explore the performance of the DoD-standard Internet Protocol (IP) and Transmission Control Protocol (TCP) in integrated voice/data internetwork environments. During FY 83, a set of experiments has been conducted aimed at finding efficient strategies for data-file transfer in a background of competing voice traffic and/or interactive data traffic. Preliminary results indicate that the appropriate strategies to maximize throughput are: (1) to maintain small buffers in the gateway and discard packets on overflow; and (2) to adjust the file transfer rate so that 2- to 3-percent packet loss (requiring corresponding retransmissions) is observed.

Finally, as described in Section 5, Lincoln has continued its role in EISN system coordination and experiment planning. An FY 83 Work Plan was prepared and delivered to DCEC, detailing FY 83 experiment plans and outlining future plans. Lincoln's role in system coordination included leadership of a wideband satellite network (WB SATNET) Task Force effort which has succeeded in achieving 3-Mbps operation on the channel, and has currently extended its activities to integration of WB SATNET equipment at the three MILDEP sites.

# 2. ROUTING ALGORITHM DEVELOPMENT FOR THE DSN

## 2.1 INTRODUCTION

Previous Lincoln efforts[1,2] have resulted in new mixed-media routing and multi-level-precedence (MLP) procedures directed at the dual DSN requirements of survivability and low cost. Performance advantages of new routing procedures after network damage were demonstrated during FY 82 using a modified steady-state network analysis program. Work during FY 83 has focused on:

(a) Development of a call-by-call simulator that is required to evaluate new routing and preemption procedures which cannot be modeled in the steady-state analysis program;

(b) Application of this simulator to study new routing procedures;

(c) Development of new user-level common-channel signaling (CCS) protocols to support the routing algorithms, and validating the protocols with the simulator; and

(d) Further specification of the details of routing and preemption procedures.

The call-by-call simulator currently includes blind preemption and all types of routing including Precedence Flooding (flooding only high-precedence calls), Precedence-Blocked Flooding (flooding only precedence calls that are blocked with mixed-media routing), and mixed-media routing with crankback. After the simulator had been validated by comparison with steady-state analysis results and detailed checking of call scenarios, the simulator was used to compare new routing procedures and techniques that improve service for high-precedence users under network damage. Results indicate that blocking for high-precedence users can be decreased significantly without preempting by allowing high-precedence calls to have greater routing freedom. This freedom can be obtained by allowing longer call path lengths or by using more complex routing procedures such as Precedence-Blocked Flooding. Results also indicate that the CCS bandwidth required to support routing procedures that use flooding is not excessive. Precedence-Blocked Flooding can support the projected high-precedence AUTOVON traffic for 1985 (roughly 1300 Erlangs) using standard, full-duplex 1200-baud CCS links.

This report contains a brief review of new mixed-media routing procedures and steady-state network analysis results followed by a description of the call-by-call simulator and a description of studies performed with the simulator.

## 2.2 ROUTING AND PREEMPTION PROCEDURES AND STEADY-STATE NETWORK ANALYSIS RESULTS

All routing and preemption procedures are designed specifically for mixed-media networks that include both terrestrial connectivity and satellite point-to-point or Demand-Assignment Multiple Access (DAMA) connectivity. In addition, all procedures treat satellite

and terrestrial links separately and use common-channel signaling to pass information between switches. The three classes of new routing procedures we are developing are referred to as:

(a) Mixed-Media routing

(b) Adaptive Mixed-Media routing

(c) Flooding routing.

These procedures are described in detail in References 2 and 3. Mixed-Media routing uses fixed routing tables and allows three types of call-processing rules: spill-forward control, remote earth-station querying, and single-stage crankback. Spill-forward control either blocks a call at a switch or routes the call to another switch not yet in the call path. Remote earth-station querying sends a CCS query message to an earth station to determine the status of earth station and satellite when the shortest path to a given destination is over a satellite. Single-stage crankback is similar to spill-forward control except that a call blocked at a switch is routed backwards to a previously visited switch where other outgoing links are used. Adaptive Mixed-Media routing is identical to Mixed-Media routing except that routing tables are automatically adapted when the network is damaged. Routing procedures that use flooding include pure Flooding, Precedence Flooding, and Precedence-Blocked Flooding. With pure Flooding, all calls are routed using the flooding technique. Precedence Flooding routes high-precedence calls using flooding and low-precedence calls using Mixed-Media routing. Precedence-Blocked Flooding routes high-precedence calls using flooding only if the calls are blocked after using Mixed-Media routing. Low-precedence calls are routed using Mixed-Media routing.

Three types of preemption procedures are being evaluated. Blind preemption, as used in AUTOVON, blindly preempts on links as a call is being set up. It may preempt an excessive number of calls on a multi-link path, and may preempt lower-precedence calls even if a call path isn't established. Source-Destination preemption, originally proposed by GTE,[4] tries to preempt a lower-precedence call that is already in progress to the desired destination switch. It does not guarantee that the path established will be the shortest path, and it may preempt an excessive number of calls on multi-link paths if no lower-precedence call is in progress to the destination. Guided preemption, which we introduced,[2] preempts the fewest calls on the shortest path to the destination.

Mixed-Media routing with two call-processing rules (spill-forward control, and remote earth-station querying) and Adaptive Mixed-Media routing were evaluated during FY 82 using a modified steady-state network analysis program. Results were presented in a Lincoln Technical Report,[3] in a conference paper,[5] and in the FY 82 Annual Report.[2] These results demonstrate that new algorithms provide significant performance improvements over algorithms such as modified forward routing (MFR) over a broad range of network conditions. In particular, the new algorithms greatly reduced the incidence of high blocking probability between node pairs, giving calls a better chance to connect after network damage. The comparisons performed were limited by the capabilities of the model used in the steady-state

network analysis program. In particular, this model cannot support multiple-call precedence levels, preemption, or routing procedures that use flooding and crankback. In addition, it does not provide information on network dynamics, CCS usage, or call setup times.

## 2.3 CALL-BY-CALL SIMULATOR DEVELOPMENT

A new call-by-call simulator was developed during FY 82 and FY 83 in order to:

(a) Evaluate all routing procedures, including those that incorporate flooding and crankback;

(b) Examine the effect of preemption and multi-level precedence features;

(c) Test CCS user-level protocols using link-level protocols compatible with CCITT No. 7;

(d) Measure CCS traffic;

(e) Examine network dynamics; and

(f) Measure call setup times.

The simulator currently includes blind preemption and all types of routing including Precedence Flooding, Precedence-Blocked Flooding, and Mixed-Media routing with crankback.

The simulator is written in a modern structured language called RATFOR which is automatically translated into portable FORTRAN code. To date, over 200 pages of RATFOR code have been written, and the simulator has been tested extensively. The simulator has been run using IBM, FORTRAN IV on an IBM 3081-D16 computer and on an Amdahl 470 computer. Operating systems used for these runs were the VM/CMS time-sharing system and the OS/VS1 batch system. The simulator has also been run under the UNIX time-sharing operating system on a VAX 11/780 computer using the newest FORTRAN release, FORTRAN 77. UNIX is currently being used for program development and documentation because it provides extensive software development and documentation tools. The OS/VS1 operating system is being used for batch runs, and the VM/CMS system is used for short test runs and to create graphics output.

A block diagram of the simulator is presented in Figure 1 where input files are indicated on the left, and output files are shown on the right. Input files are compatible with those of existing DCA steady-state network analysis programs. They contain network controls, network topology information, trunk group sizes, and offered traffic information.

Network controls for a simulation run are altered by editing a file that defines values of control parameters. The routing procedure and the type of preemption can be selected for each of five call precedence levels. The routing procedure can be Forward Routing (FR), Modified Forward Routing (MFR), or any of the new types of routing procedures. When blind preemption is desired, it can use a friendly or ruthless preemption search algorithm at each switch to decide when to preempt. Switches using the ruthless preemption algorithm examine outgoing links, as indicated in the routing table, and use the first link with a free

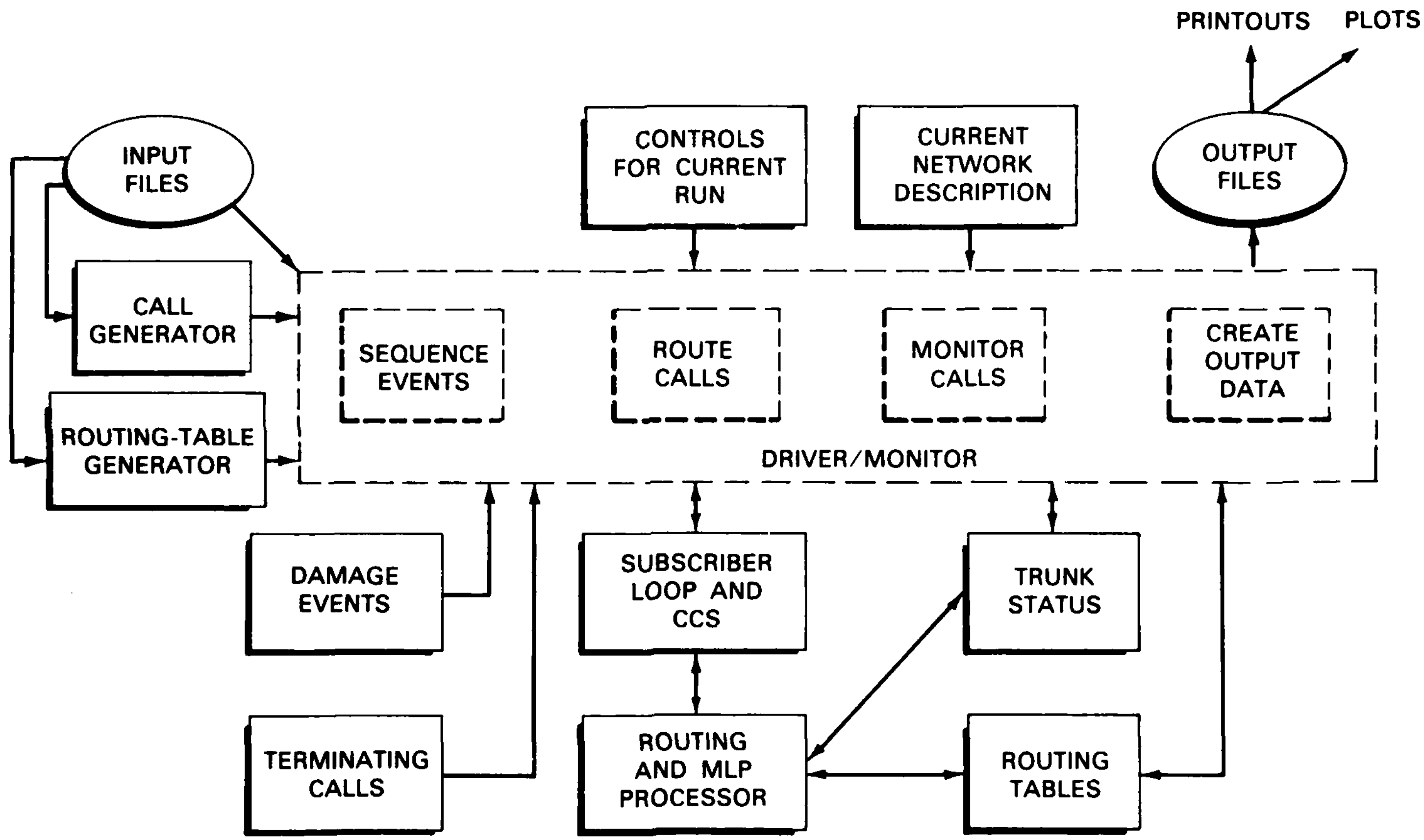



**Figure 1. Block diagram of call-by-call simulator.**

116997-N-05

or preemptable trunk. Switches using a friendly search algorithm first examine outgoing links, as indicated in the routing table, looking for a free trunk and treating preemptable trunks as busy. If no free trunk is found after examining all links, then links are examined a second time and the first preemptable trunk found is used. Other network controls that can be selected for each precedence level are the maximum number of links in call paths, the maximum number of routing-table entries to search, the maximum number of satellites in call paths, and the maximum number of crankbacks in a call path. The type, amount, and time of damage can also be selected as well as the simulation run time and the time intervals for plots, details, and statistical printouts.

Two input files are created before simulation runs. One file, created by a call generation program, contains a list of offered calls. Another, created by a routing-table generation program, contains routing tables for all nodes in a network. The random-number generation algorithm used in the call generation program is identical to that used in the RCP. This greatly simplifies RCP validation because identical patterns of offered calls can be produced both in the simulator and in RCPs without transferring large call files between systems.

Output files from the simulator contain detailed statistical printouts in a format that is similar to the format used by DCA steady-state network analysis programs. In addition, some files are used by a graphics software system called TELLAGRAF to automatically produce plots of network parameters vs time and a plot containing the histogram of blocking probabilities experienced by each point-to-point pair in the network. Parameters that are plotted vs time include: total blocking probability, blocking probability by precedence level, calls preempted, call path length in links, CCS bits transmitted, calls cranked back, and calls that queried a remote earth station.

Information on both interval and overall statistics is contained in output printouts. Interval statistics are collected during short (e.g., 3 min.) intervals for each precedence level. These statistics contain information on blocking probability, calls preempted and preempting, CCS bits transmitted, call path length, call setup time, reason for call blocking, number of crankbacks, number of calls flooded, and blocking probability after flooding. Overall statistics are collected from the time the simulation stabilizes to the end of a run, or before and after damage. These statistics contain all the information in interval statistics plus other statistics, again by precedence level. Additional statistics include: point-to-point blocking between all node pairs, blocking and occupancy for each trunk group, blocking and traffic to and from all nodes, blocking and setup time by call path length, and statistics based on a histogram of blocking probabilities between all node pairs.

The main control section of the simulator, labeled DRIVER/MONITOR in Figure 1, reads in and validates the input files and then reads in offered calls one at a time and offers them to the network. It also initiates damage, terminates calls, monitors calls, and creates output printout and graphics files. The two most complex components of the simulator are the "Subscriber Loop and CCS" module and the "Routing and MLP Processor"

module. The "Subscriber Loop and CCS" module behaves like a subscriber loop on an originating or terminating switch, and like a CCS network between tandem switches. It transports CCS messages, keeps statistics on CCS traffic, computes CCS transit time, updates statistics, prints out detailed information on simulator operation when desired, and transports subscriber loop signals (new call, phone ringing, phone answered, phone busy) between telephones and the originating and terminating switches. This module also tells the "Routing and MLP Processor" the identity of the switch it is to simulate when it is activated.

The "Routing and MLP Processor" module performs the routing, preemption, and multi-level precedence functions that are necessary in a switch. Software for this module is meant to serve as a model for software used in a real switch. This module receives a CCS message or a subscriber loop signal, takes the appropriate local action, and then, if necessary, outputs one or more new CCS messages. To do this, it implements new user-level CCS protocols compatible with CCITT No. 7 that support all routing and preemption procedures. These tasks are performed using routing tables and information describing the status of trunks, the current network topology, and current calls in progress. The simulator has been thoroughly validated and tested for all types of routing and preemption. Validation first involved examining detailed simulator behavior using both large and small networks and patterns of offered calls that exercised critical routing and preemption code. This was facilitated by control variables that varied the amount of details printed, and that turned detailed printouts on either for a complete run or during a specific time interval within a run. In addition, printouts of statistics were examined for consistency, and inconsistency checks were placed throughout the program. Finally, simulation results were compared with predictions obtained using the modified DCEC steady-state network analysis program and with analytic predictions. Comparisons were performed with 2-, 4-, and 20-node networks under normal load, with damage, and with blind preemption and overload using from 100 to 23,000 simulated calls per comparison. The average point-to-point blocking from the simulator ranged from 0.0035 to 0.759 with a standard deviation, estimated from interval statistics, ranging from 0.001 to 0.025. The overall blocking probability obtained with the simulator was not significantly different from the predicted blocking. The difference between the simulator and the predicted blocking ranged from 0.003 to 0.025 and only one of fifteen differences was statistically significant at the 0.05 level (students' test).

## 2.4 CALL-BY-CALL SIMULATOR PERFORMANCE RESULTS

Two series of simulator runs were performed to compare new routing procedures and multi-level precedence procedures after network damage. The first series of runs was performed before flooding had been added to the simulator, and the second series was performed with Precedence Flooding and Precedence-Blocked Flooding. Both series of runs were performed with 20-node network DSN1 under normal conditions and with the satellite destroyed.

Link and switch locations for network DSN1 are presented in Figure 2. Solid lines in this figure represent land links, and dashed lines represent links to one DAMA satellite.

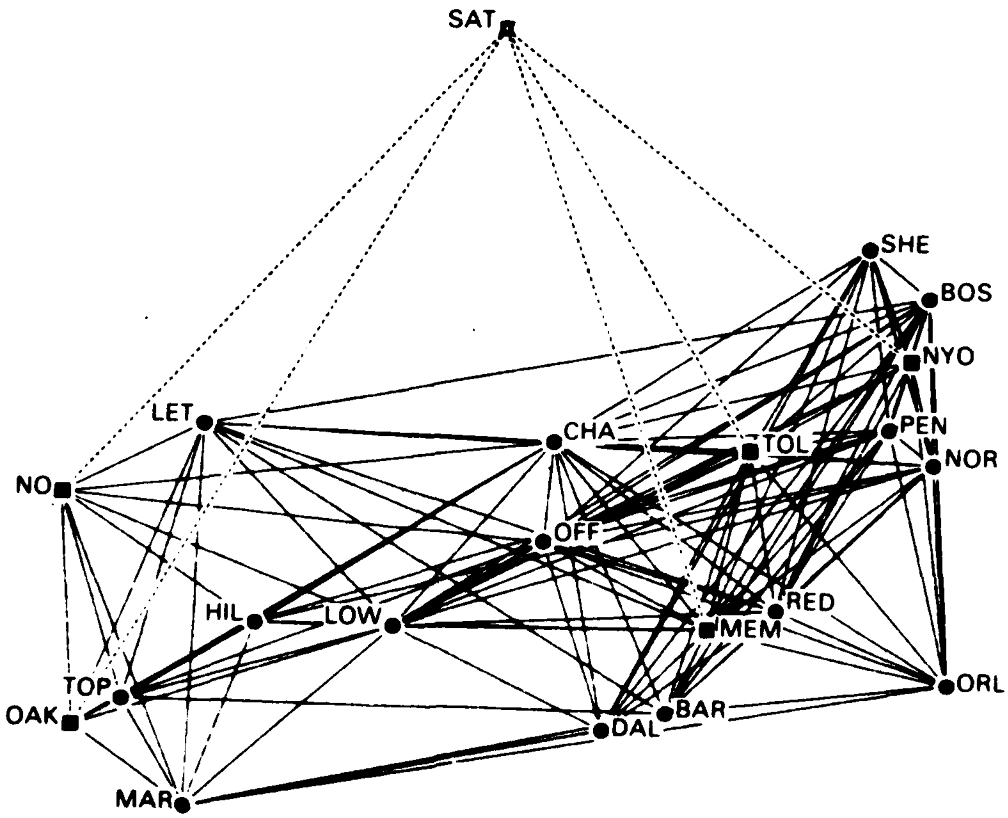



**Figure 2.** Test network DSN1.

Network DSN1 is a minimum-cost network designed for a link-blocking probability of 0.1 and designed to route roughly 1/3 of all traffic over the satellite under normal conditions. It includes 20 switches, one DAMA satellite, and five earth stations. The total traffic offered to this network was 1450 Erlangs. Roughly 20 percent of this traffic consisted of Priority calls and the remaining 80 percent consisted of Routine calls.

During the first series of simulation runs, roughly 30,000 calls were offered per run. Low-precedence calls were routed using spill-forward Mixed-Media routing. These calls could travel one link more than the number of links in the shortest path to each destination. High-precedence calls were routed:

(a) The same [illegible] calls with spill-forward Mixed-Media routing,

(b) Allowing t[illegible] links instead of one in call paths,

(c) The same as [illegible] with crankback,

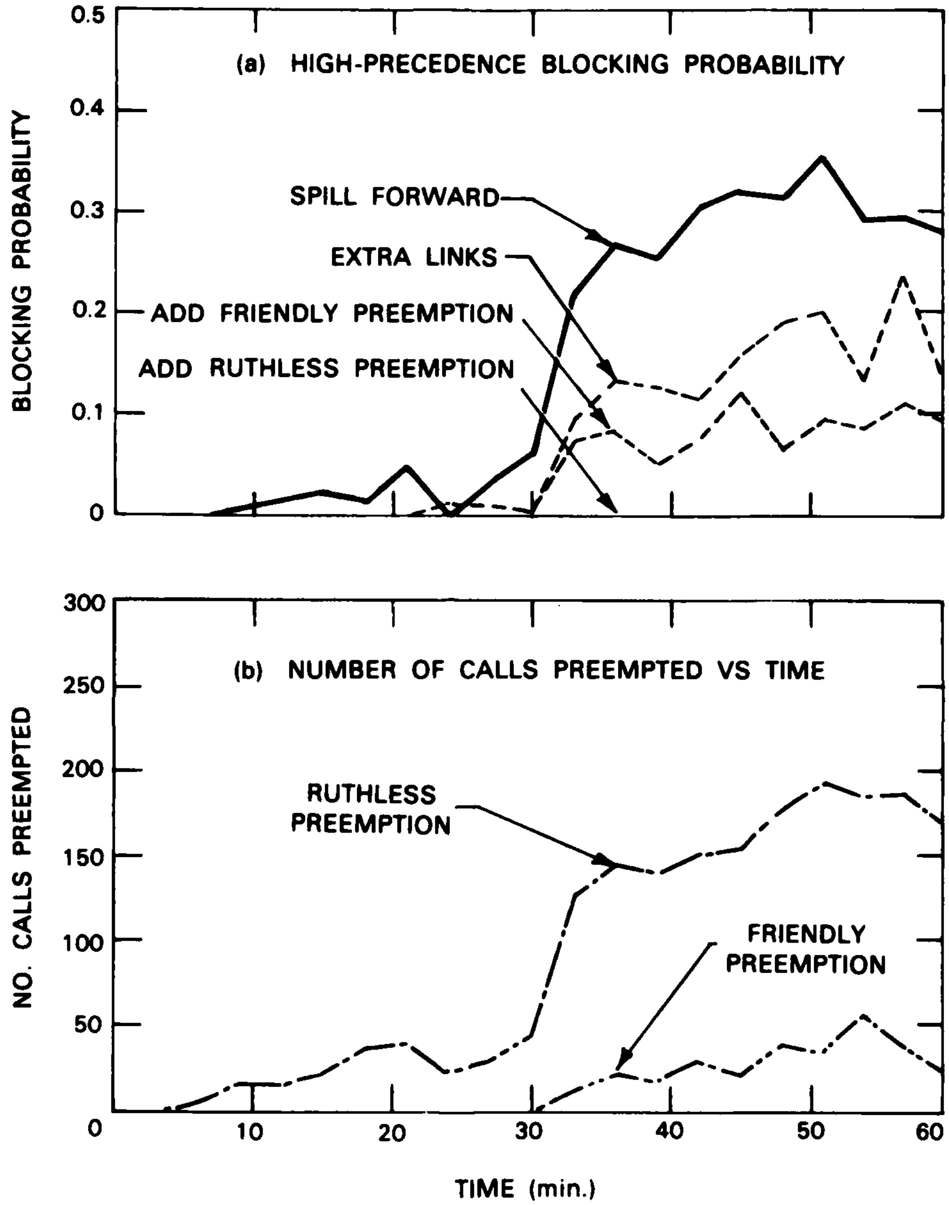



**Figure 3(a-b).** Example of effects of network damage on blocking and preemption in network DSN1 (satellite destroyed after 30 min.).

(d) The same as (b) but with friendly preemption, and

(e) The same as (b) but with ruthless blind preemption.

Simulation runs typically used 6 min. of processing time on an Amdahl 470 computer to simulate 60 min. of real time. Figure 3(a-b) presents sample simulator output plots produced when the satellite was destroyed after 30 min. of simulated time. Figure 3(a) contains curves of blocking probability vs time for high-precedence calls. A curve is not plotted for crankback because results obtained with crankback and extra links were similar. Blocking for low-precedence calls is not plotted because it varied little as different techniques were used to route high-precedence calls and was near the top curve in Figure 3(a).

Figure 3(a) indicates that blocking is low before network damage and then rises after damage for all conditions except ruthless preemption. Figure 3(b) indicates that the number of calls preempted is low before damage and then high after damage with ruthless preemption. Under this condition, roughly 18 percent of all low-precedence calls that reached the talking stage were preempted. Although this figure appears to indicate that ruthless preemption is the best choice in terms of reducing high-precedence blocking, it also illustrates that it is possible to reduce high-precedence blocking without preempting calls. Permitting high-precedence calls to have longer path lengths reduced high-precedence blocking from 0.28 to 0.15 without preempting and with little effect on low-precedence blocking. The minimal advantage of crankback in the above runs was due to path-length limitations that disallowed the long call paths created with crankback. Further runs are planned to examine crankback when longer paths are allowed.

A second series of simulator runs was performed after Precedence Flooding and Precedence-Blocked Flooding were added to the simulator to evaluate these new procedures. All runs were again performed with 20-node network DSN1 under normal conditions and with the satellite destroyed using roughly 30,000 offered calls per run. Three types of routing that incorporate flooding were examined. In the simplest (Flood All), all calls were routed using flooding. In the others, either all high-precedence calls were routed using flooding (Precedence Flooding), or only high-precedence calls that were blocked after being routed with spill-forward Mixed-Media routing were routed using flooding (Precedence-Blocked Flooding). Processing time to simulate 1 h of real time on an Amdahl 470 computer ranged from 8 min. with Precedence-Blocked Flooding, to 58 min. when all calls were flooded.

Figures 4 and 5 compare results obtained after damage with flooding procedures to results obtained with: primary path routing, spill-forward Mixed-Media (SFMM) routing allowing three extra links in high-precedence call paths, and SFMM routing with blind preemption. Figure 4 presents the average blocking probability and call path length for high- and low-precedence calls and the total number of calls preempted. Figure 5 presents the average bit transmission rate on CCS links under normal conditions and after the satellite is destroyed.

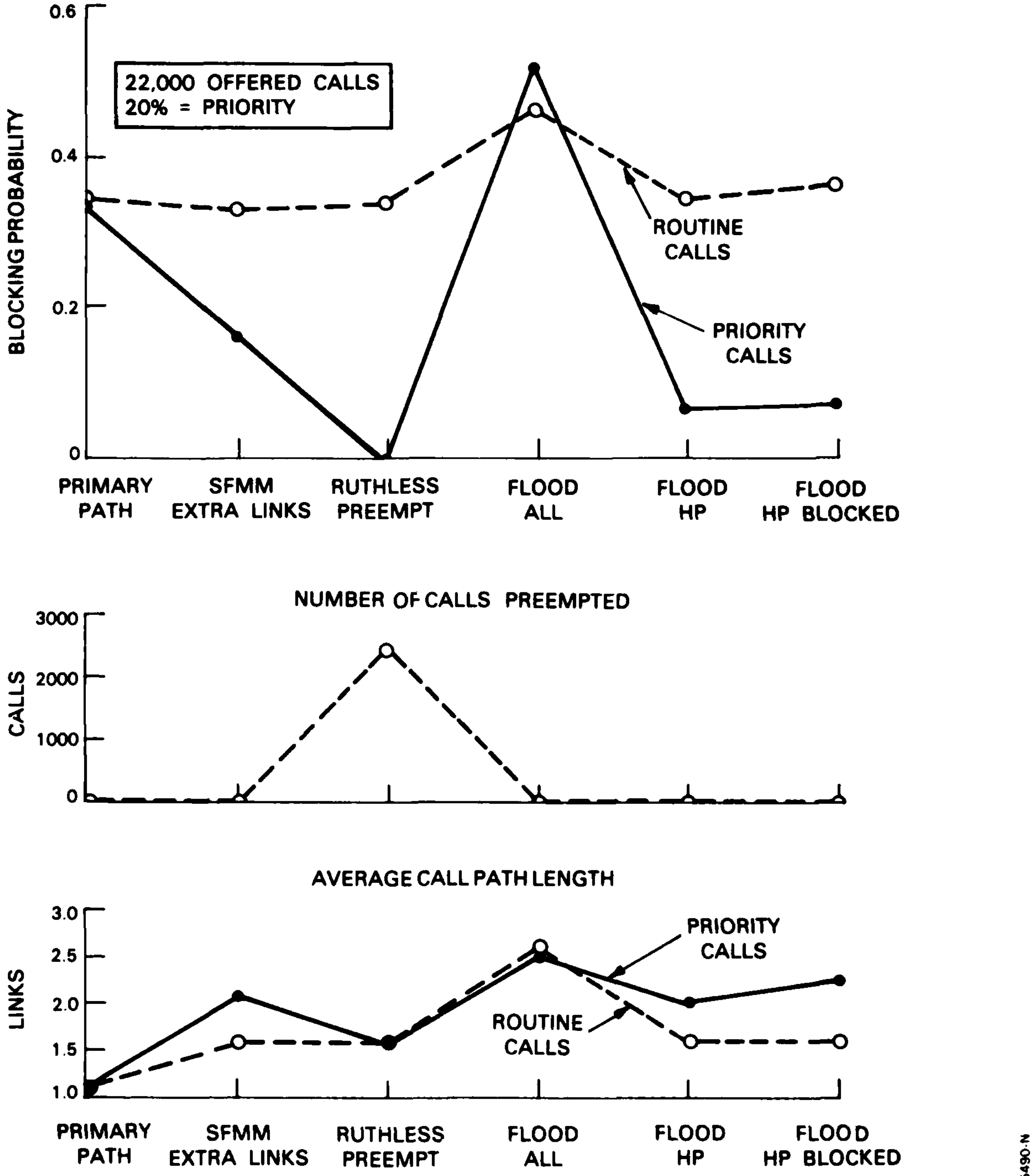



**Figure 4. Blocking probability for high- and low-precedence calls in DSN1 with satellite destroyed.**

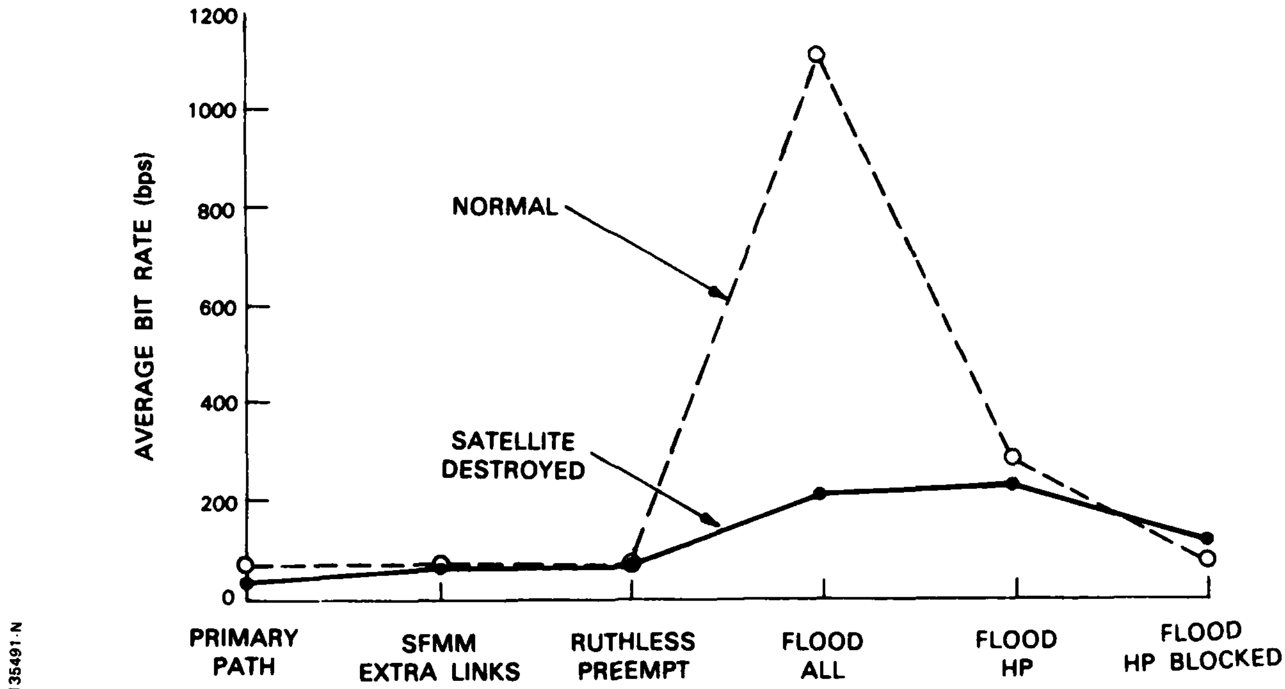



Figure 5. Average bit rate on CCS links in DSN1 with satellite destroyed.

Under normal conditions, the average point-to-point blocking was low (0.0 to 0.05). The average load on CCS links was excessively high with pure Flooding and Precedence Flooding (1100 and 250 bps), but low and near the rate required without flooding with Precedence-Blocked Flooding (73 bps).

After the satellite was destroyed, Precedence Flooding and Precedence-Blocked Flooding reduced high-precedence blocking to low values (0.066 and 0.072) without preempting. High-precedence blocking was 0 with ruthless preemption and 0.15 when high-precedence calls were allowed to traverse more links in call paths. In addition, the average CCS transmission rate when high-precedence calls were routed using Precedence-Blocked Flooding was found to be low and less than twice the rate required with Mixed-Media routing (115 vs 64 bps). These results demonstrate that Precedence-Blocked Flooding can provide good service to high-precedence calls without preemption and without requiring excessive CCS communication bandwidth. In addition, they suggest that Precedence-Blocked Flooding should be supplemented by preempting to complete the residual high-precedence calls which do not find a route via flooding. This would provide good service to high-precedence calls but preempt fewer calls than blind preemption.

Current plans for the simulator are to further compare flooding and preemption techniques in networks with severe damage, traffic overloads, and more limited connectivity than

network DSN1. Further simulator development is also planned to add source-destination and guided preemption and to add a more complex model of call retry behavior after a call is blocked or preempted. In addition, the simulator will continue to be used as a basis for implementing routing algorithms and CCS communication protocols in the RCP for experimental EISN tests.

# 3. EISN INSTRUMENTATION AND INTEGRATION

The purpose of the EISN system is to provide a system-level test bed for the evaluation of advanced communications networking techniques, including survivable network routing algorithms using a mix of transmission media, for application in the DSN. As illustrated in Figure 6, EISN is being developed in phases. The interim routing/control experimental facility has supported basic experiments in satellite/terrestrial integration, alternate routing, and data communication. The advanced facility currently under development includes off-the-shelf digital switches and flexible outboard RCPs to allow experimental test of new routing and preemption algorithms in a test bed which has key DSN features including: digital switches, flexible access to a mix of transmission media, and interswitch CCS communication. Two conference papers on the experimental system were prepared during FY 83. An overview of the EISN system is given in Reference 6. A description of experiments on the wideband system conducted under both DCA and DARPA sponsorship is given in Reference 7.

Lincoln has played a major role in the development and integration of experimental subsystems to support the EISN experiments. Major Lincoln developments prior to FY 83 were Packet/Circuit Interface (PCI), the Telephone Office Emulator (TOE), and the Internet Packet Gateway (IPG). During FY 83, PCIs and TOEs were installed at three new EISN sites, and their capabilities were extended to include precedence and preemption. The capabilities of the IPG and associated Measurement Host (MH) equipment were increased (see Section 4) to support data protocol and voice/data integration experiments.

However, the major FY 83 Lincoln effort in EISN instrumentation and integration has been the detailed design and development of the RCP/switch facility. As described below, two RCP/switch systems are currently operational at Lincoln, and the capability to control local calls and to accommodate precedence features has been demonstrated. The RCP and switch hardware are illustrated in Figure 7, which shows a United Technologies LEXAR UTX-1200 switch and a PDP-11/44-based RCP.

## 3.1 PACKET/CIRCUIT INTERFACE AND TELEPHONE OFFICE EMULATOR

The function of the Packet/Circuit Interface (PCI) at an EISN node is to provide an interface between analog, circuit-switched telephone voice and signaling and their counterparts in the packet-switched wideband satellite network. The TOE was designed to function as a very small class 4/5 telephone central office to provide traffic for the initial experiments with the network of PCIs. As the source of traffic is phased over to the telephone switches associated with the RCPs, the TOEs will continue to serve as test equipment. In addition, portions of the TOE hardware will continue to be used by the PCI to make connections to external telephone equipment. In FY 83, the final three field installations of the PCI and TOE were accomplished, bringing the total to five sites. Development of PCI software continued, notably the addition of precedence levels for calls and a preemption capability. In addition, a number of steps were taken to prepare the PCIs and TOEs to be integrated into the coming network of RCPs. The following sections describe this work in more detail.

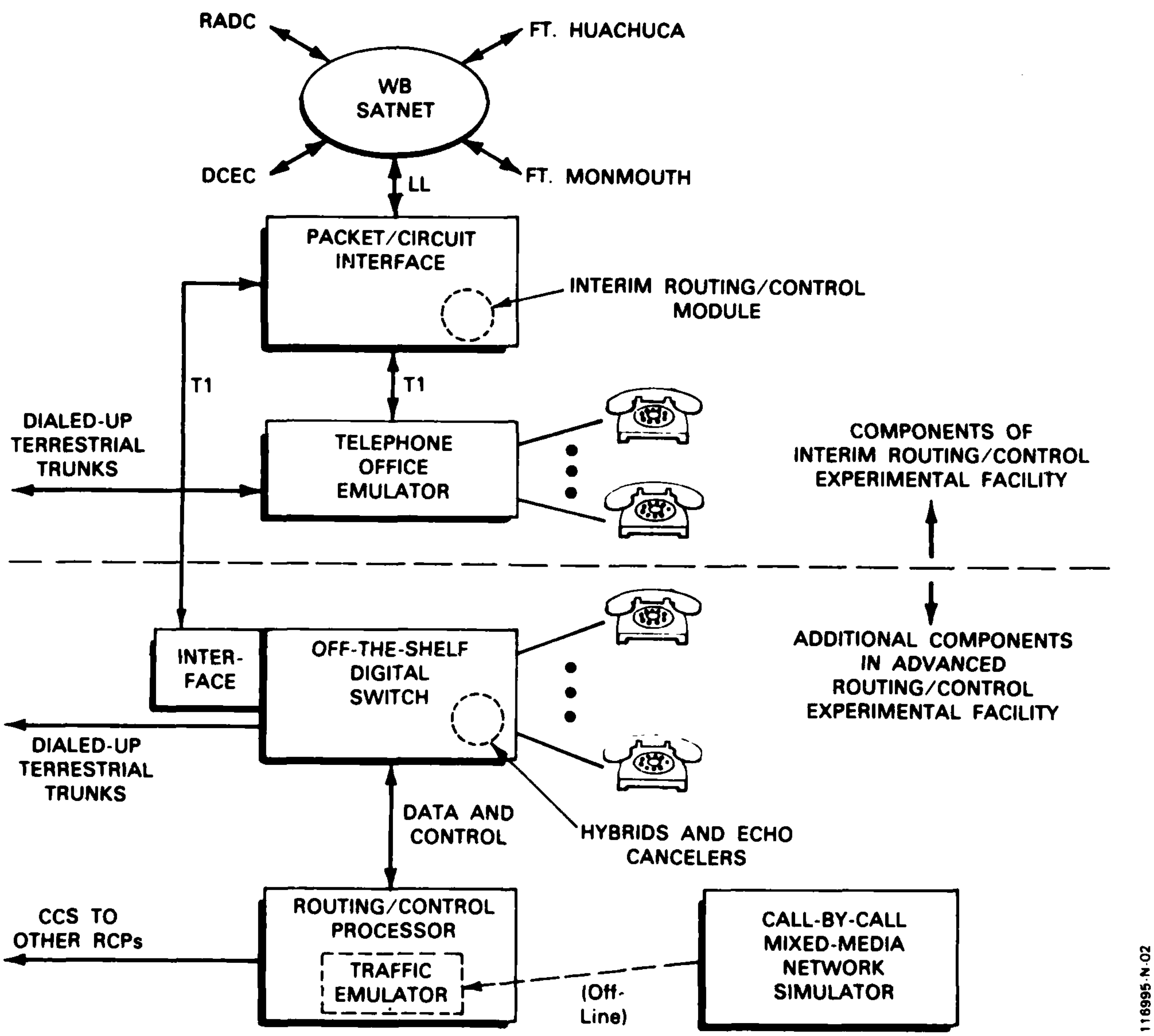
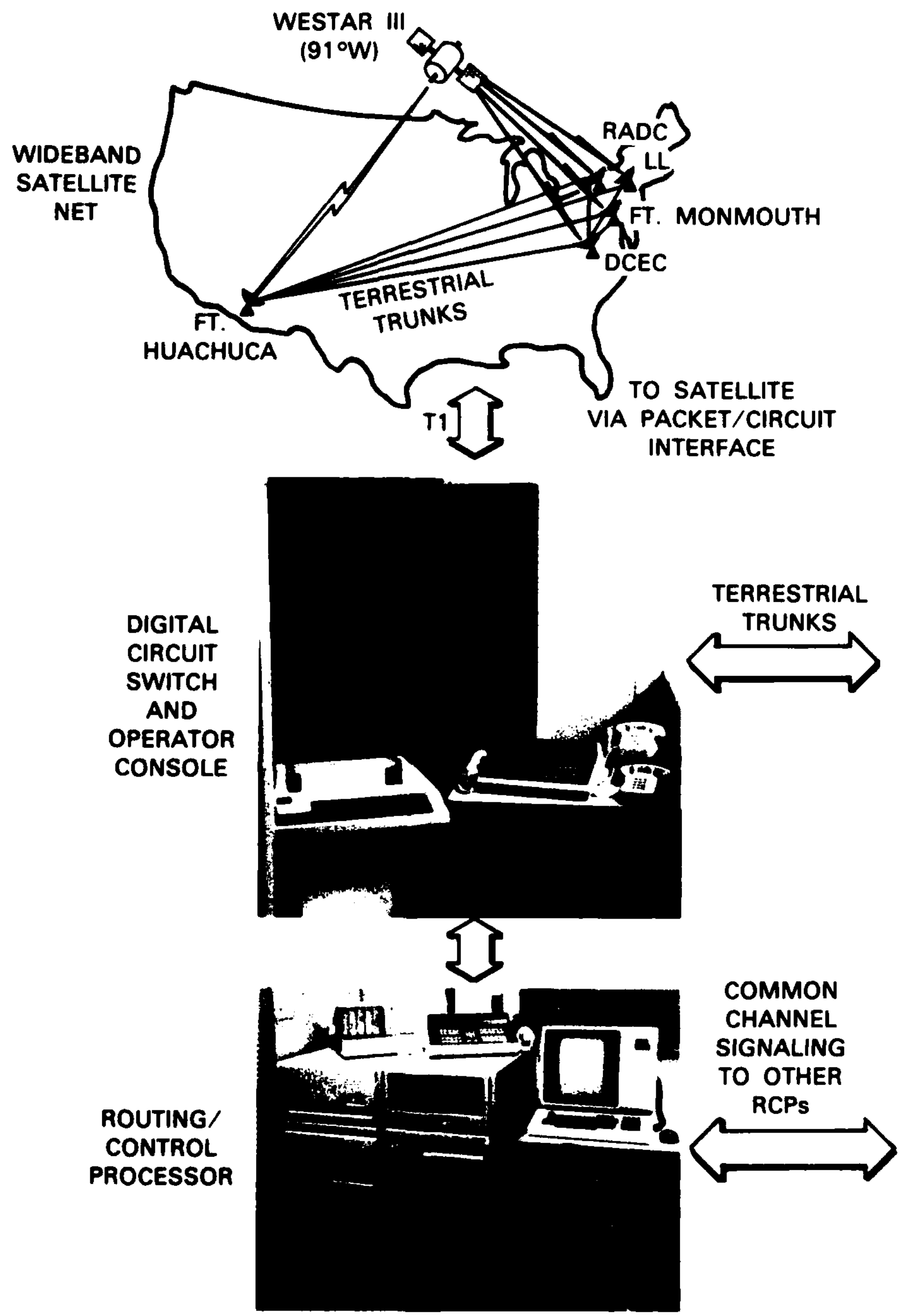
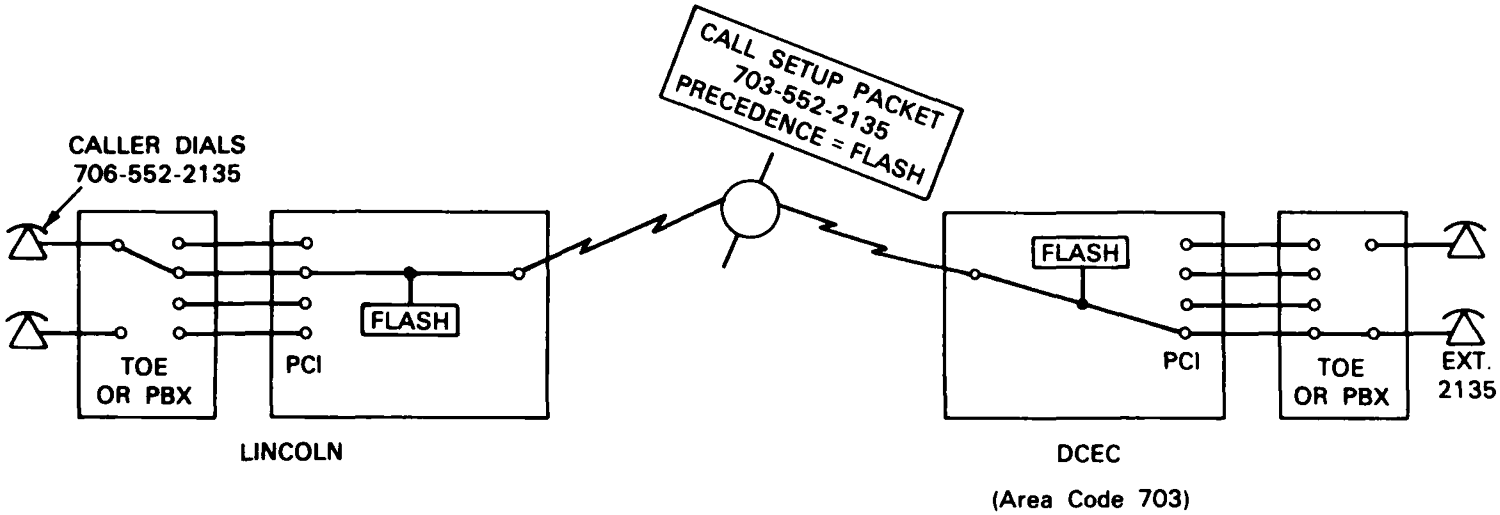



Figure 6. Advanced EISN experimental facility including digital switch and RCP.




126006-N-01

**Figure 7. Experimental switch and RCP facility including United Technologies LEXAR UTX-1200 digital switch and PDP-11/44-based RCP.**




| DIALED AREA CODE | PRECEDENCE LEVEL |
|---|---|
| 703 | ROUTINE |
| 704 | PRIORITY |
| 705 | IMMEDIATE |
| 706 | FLASH |
| 707 | FLASH OVERRIDE |

**Figure 8. Transmission of precedence among PCIs.**

135492 N

### 3.1.1 Field Installation of PCI/TOE Equipment

In FY 83, installations of PCI/TOE equipment were made at Rome Air Development Center (RADC), New York (January); Fort Huachuca, Arizona (June); and Fort Monmouth, New Jersey (September). Funding for the construction and installation of this equipment was provided by RADC for the Rome, New York site and by the U.S. Army Communications Systems Agency for the two Army sites. PCI/TOE equipment is now operating at all five EISN sites, the first two being Lincoln Laboratory and the Defense Communications Engineering Center. Capabilities that were successfully tested at each new site included:

(a) Calls between two TOE phones.

(b) Calls from the PCI through the site's PBX.

(c) Calls from the PCI looped through the site's PSAT.

(d) Remote control of the PCI/TOE equipment via a telephone modem connection to Lincoln Laboratory, including the ability to download new versions of software for the PCI's processors.

Communication over the wideband satellite network (WB SATNET) from the three new sites will become possible upon the installation of an ESI (Earth Station Interface, a burst modem) at each site.

### 3.1.2 Incorporation of Preemption into PCI Network

The program in the PCI's UMC-Z80 processor was augmented to support both assignment of precedence levels to calls and preemption by calls with higher precedence levels. This preemption scheme is very simple and will be replaced gradually by the more sophisticated preemption scheme being implemented in the RCPs. However, it will be able to operate compatibly with the RCP preemption scheme during the time when some sites have a PCI but no RCP.

Figure 8 shows how a caller can choose higher precedence by augmenting the area code of the dialed number. For example, the normal area code for the DCEC site is 703. Under the precedence scheme adopted, the use of the normal area code results in an assignment of ROUTINE precedence. In Figure 8, the caller dialed 706. The PCI router interpreted that as a FLASH call to DCEC and sent a call set-up packet with that information. Figure 9(a-c) shows how the PCIs perform preemption. In Figure 9(a), a FLASH call is entering the left PCI, and its router determines that no more satellite capacity is available to it. Therefore, it finds a call of lower precedence on the satellite and decides to preempt it. Figure 9(b) shows how preemption is accomplished. The local party to the preempted call is connected to a tone generator to notify him that his call has been preempted. A call take-down packet with a PREEMPT field is sent to the distant PCI, which applies a tone to the other end of the preempted call. Meanwhile, a call set-up packet seizes the newly available satellite capacity for the FLASH call, which is shown connected in Figure 9(c).

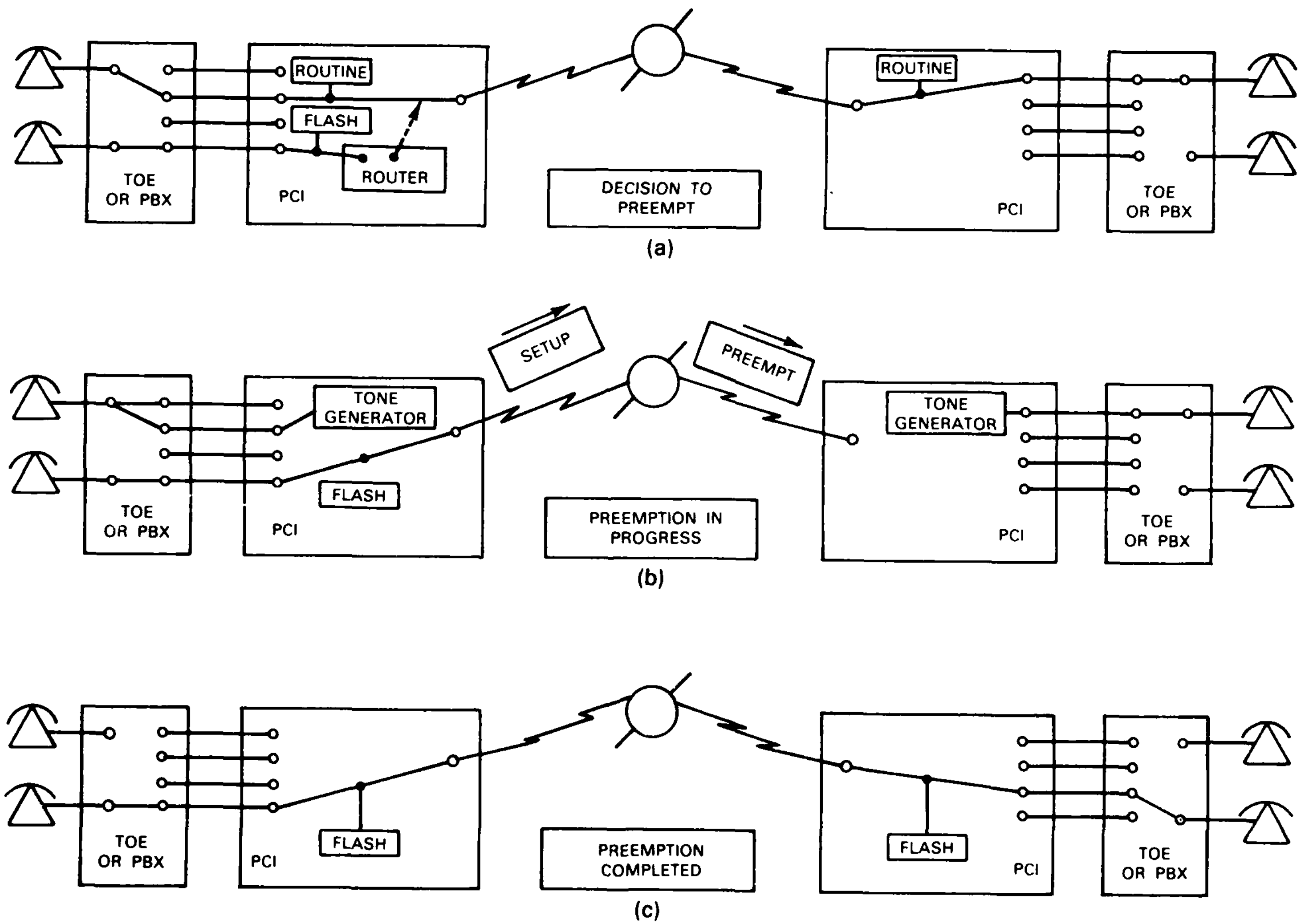



Figure 9. Execution of preemption by PCIs.

135493-N

### 3.1.3 Steps Toward Phasing PCIs into RCP Network

As the advanced EISN routing facility is developed, the TOE will be replaced as a traffic source at each EISN site by a class 4/5 commercial telephone switch such as the United Technologies LEXAR UTX-1200 or the Northern Telecom SL-1. (The digital channel banks and echo cancelers in the TOE cabinet will remain in use.) The simple routing and preemption capabilities of the PCI will be superseded by those of a RCP, which will function as an outboard controller for the commercial telephone switch. (The PCI will continue in service as a forwarder of speech and signaling over the WB SATNET.) There will be a phasing-in period during which some EISN sites will have both a PCI and an RCP, while others will have just a PCI. Steps have been taken in the construction of the PCIs and TOEs to allow sites with only a PCI to operate compatibly with the fully equipped sites and to minimize the changes to a PCI and TOE that must be made when a RCP is installed at that site. These steps are grouped below into two categories, associated respectively with the voice interface to the telephone switch and with the signaling interface to the RCP.

*Voice Interface with Telephone Switch:*— The telephone connections to the PCI/TOE without and with a RCP are shown in Figures 10 and 11, respectively. Each figure shows six voice connections, of which only four at a time may be selected (because of processing and memory limitations in the UMC-Z80 processor of the PCI). In Figure 10, two voice connections are to TOE phones (special Lincoln-built phones with separate paths for each direction of voice and for each direction of signaling), one is to an ordinary 2-wire subscriber extension on the site's PBX, and the fourth is to a 4-wire E&M trunk termination (where available) at the site's PBX. The 2-wire connection enables telephones on the site's PBX to receive calls through EISN, while the 4-wire E&M connection allows both the receipt and the origination of EISN calls.

The design of the TOE's wiring and the inclusion of a four-position echo-canceler chassis in the installation of PCI/TOE equipment during FY 83 will allow a simple changeover to the connections of Figure 11 when an RCP is installed later. The only changes needed will be the replacement of one plug-in channel unit with another, the wiring of several jumpers on the echo-canceler chassis, a table entry in the software of the PCI's UMC-Z80 processor, and some switch settings in the PCI. After these changes, the normal connection would be four 4-wire E&M trunks to the RCP's phone switch. However, it will sometimes be desired, for testing purposes, to replace two of the 4-wire E&M trunks by the two TOE phones. This change can be done with just switch settings and a software table entry. No wiring changes will be needed.

*Signaling Interface with the RCP:*— When an EISN site receives its RCP, the role of the PCI will be downgraded, but it still will have two main functions:

(1) to forward CCS messages over the WB SATNET between RCPs, and

(2) to create point-to-point packet-voice connections over the WB SATNET between the phone switches of two RCPs.

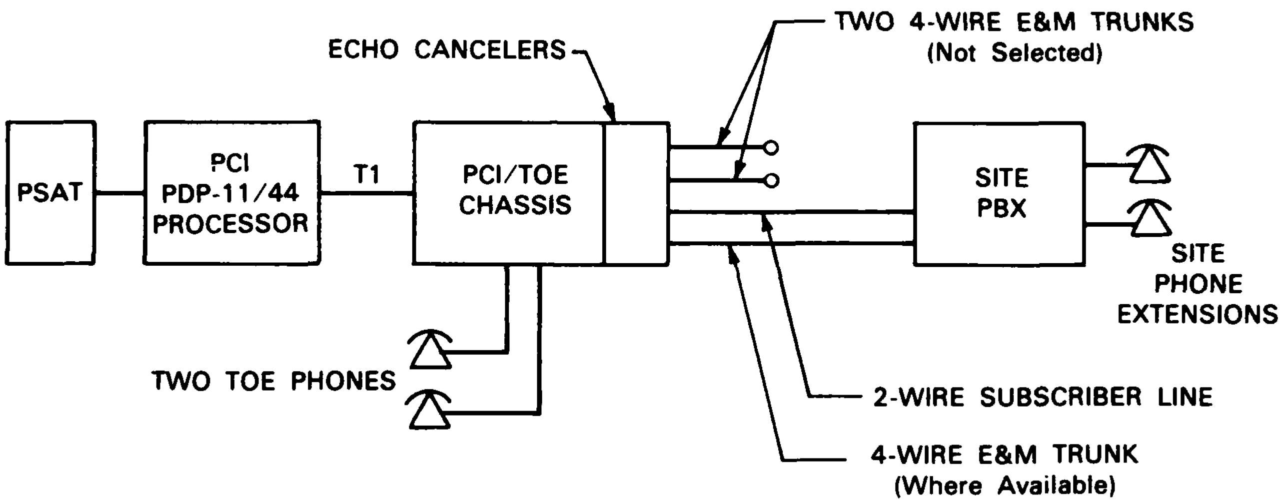



**Figure 10. Telephone connections to PCI/TOE at EISN site without a RCP.**

135494-N

135495-N

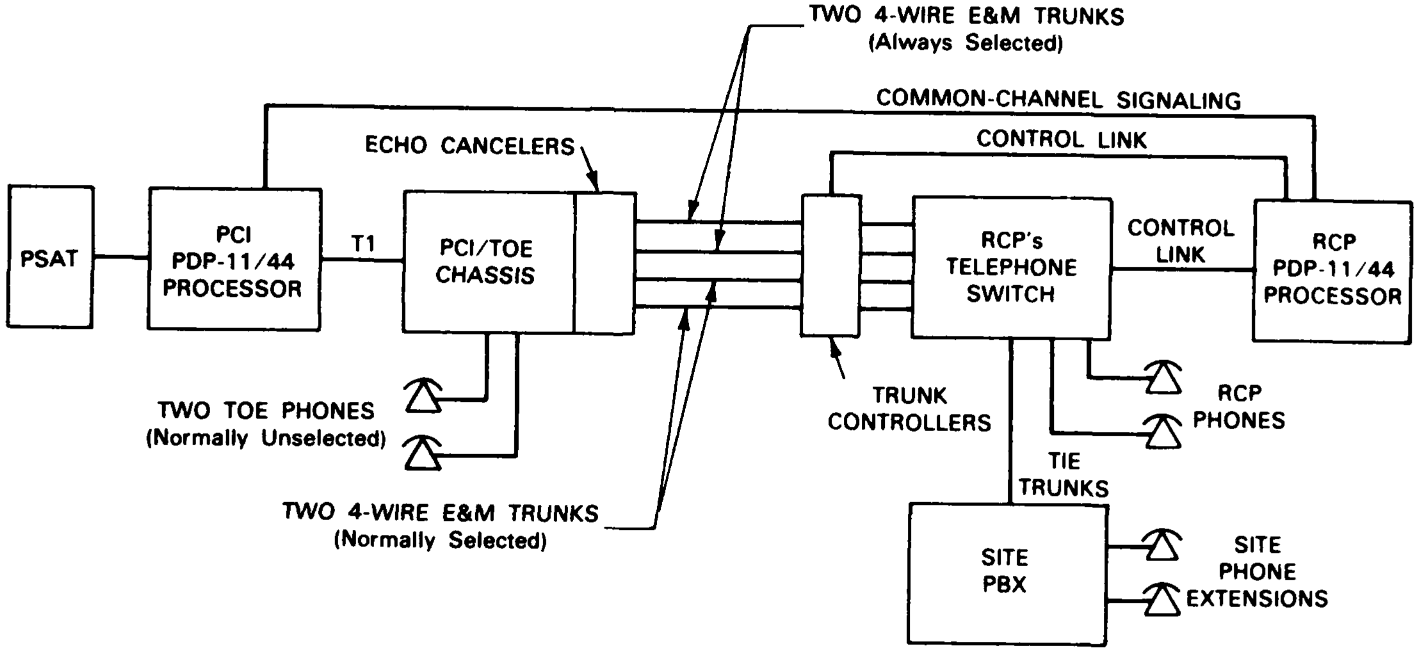



Figure 11. Telephone connections to PCI/TOE at EISN site with a RCP.

In addition, the PCI will be temporarily the only means a RCP has of reaching any destination at distant sites without RCPs. It will remain permanently the only means of reaching its TOE phones or (at certain sites) LEXNET phones on local-area packet-voice networks (LEXNETs).

The basic rule for signaling compatibility between an EISN site with a RCP and a PCI and one with just a PCI has two parts:

(1) The PCI at the site with no RCP should not have to be aware that RCPs exist; i.e., it should signal only to the PCI at the advanced site in the same way it always had before the RCP was installed. This means that the PCI software at site A need not change when a RCP is installed at site B.

(2) The RCP, when it extends a call to its local PCI for forwarding to a site it knows has no RCP, must give up the rich signaling environment it shares with other RCPs and be content to exchange with its local PCI a signaling repertoire functionally equivalent (though different in format) to the signaling between PCIs at sites having no RCPs. For example, the RCP can expect the precedence level of a call to be transmitted to the distant PCI, because the PCI network supports preemption. It cannot expect the distant PCI to tandem-switch the call to a third site, because the PCI routing algorithm does not support tandem switching.

The actual signaling between a RCP and its local PCI will take place over an RS232 link at 9600 baud between a port on the RCP and a Serial I/O chip on the PCI's UMC-Z80 processor. The hardware for this link has been installed on the Lincoln Laboratory and Ft. Monmouth PCIs, and will be retrofitted to the other sites when their RCPs are installed.

The signaling is compatible with CCITT No. 7 signaling in that it uses the same message format. The user part is unique to the RCP world, reflecting the richer signaling repertoire of RCPs compared with commercial telephone networks. The basic call set-up and take-down protocols have been defined for routing hops over the WB SATNET between two RCPs via their local PCIs. In particular, the interaction between those protocols and the NVP (Network Voice Protocol) and ST (Stream Protocol) now in use on the WB SATNET has been specified.

### 3.2 DIGITAL SWITCH INTEGRATION

The FY 82 Annual Report[2] reported that detailed design efforts had begun on the advanced routing/control experimental facility, including commercial switches and outboard RCPs, and noted that selection of suitable switches had been made for the Lincoln and DCEC sites. These switches have been procured, with some minor modifications as noted

below. Development of the RCP, including its interfaces to the switch, is described in the next section.

The Lincoln switch was received early in FY 83, and was installed by a Stromberg-Carlson field installation team in accordance with standard practices for new operational facilities. Lincoln personnel carried out further consultations with Stromberg-Carlson engineers on the detailed design of the RCP/switch interface, and it was realized that it would be unduly difficult and expensive to modify the switch CPU as proposed last year to provide the necessary additional features, namely (1) notifying the RCP of all call completions on a specified trunk group, and (2) allowing preemption through directed call termination. It was decided instead to provide these additional features by means of custom-built signaling controllers, described in the next section, and to give the RCP functional access to the switch CPU in a manner precisely equivalent to that of an operator on a standard attendant console.

To this end, Stromberg-Carlson agreed to modify one of their standard attendant consoles by adding a set of wires and a connector giving external access to (1) the electrical signals corresponding to attendant keystrokes, and (2) the electrical signals which operate the lights and the alphanumeric display on the console. This modified console was delivered to Lincoln with the switch, in addition to an unmodified console which is to be used for conventional operator functions in support of experiments.

In mid-FY 83, the DCEC switch was received and was installed adjacent to the Lincoln switch, where it will remain through the first three quarters of FY 84. It was provided with a modified attendant console identical to that of the Lincoln switch, as well as an ummodified console. The status of RCP software and special interface hardware for both switches is discussed below.

As a matter of interest, it should be noted that Stromberg-Carlson was bought by United Technologies, Inc. during FY 83. The company name was changed to United Technologies LEXAR, Inc. (in that United Technologies already had a LEXAR division which made switches), and the designation of the Lincoln and DCEC switches was changed to UTX-1200. There were no changes in switch design and operation, or in the nature of Lincoln's contracts with the company.

## 3.3 ROUTING/CONTROL PROCESSOR DEVELOPMENT

### 3.3.1 Overview

The major focus of FY 83 work on the RCPs was to put together two operational RCP/switch systems. This involved: integrating hardware and software RCP/switch components, developing custom interfaces used by the RCP to monitor and control the switch, design and coding of software to support local calls, and design of software to support inter-switch calls. Two RCP/switch systems are currently operational at Lincoln Laboratory,

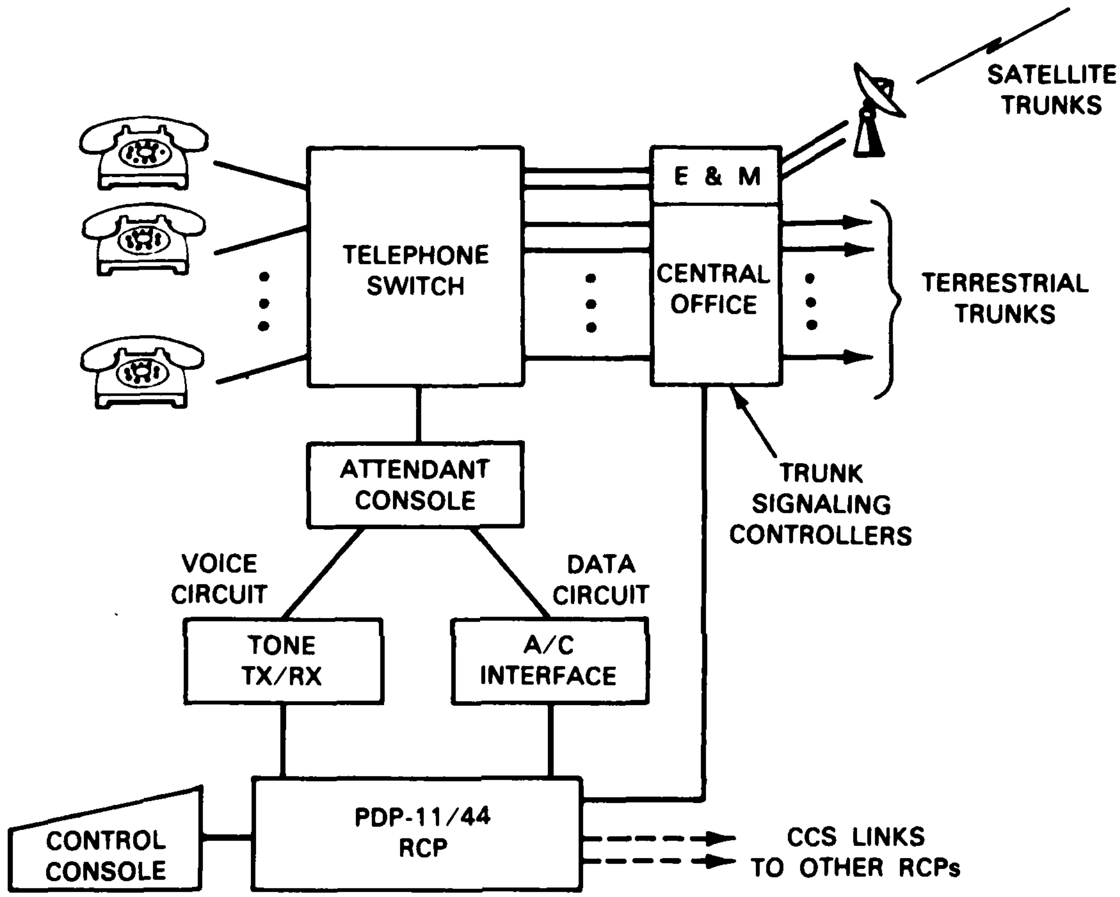



**Figure 12. RCP/switch configuration showing interfaces between RCP and switch.**

and software to set up local calls via RCPs on these systems has been demonstrated. This software allows a call to be placed from any phone on the experimental UTX-1200 commercial telephone switch to any other phone on that switch or on the Lincoln Laboratory SL1 PBX switch. In addition, when a higher-precedence call is attempting to connect, callers in an ongoing conversation are notified by an intermittent tone and can hang up and be connected to the higher-precedence call.

### 3.3.2 Hardware

A block diagram of a RCP/switch system is presented in Figure 12. Each system includes a UTX-1200 commercial telephone switch, a PDP-11/44 computer and peripherals, and custom interfaces used to control and monitor the switch. All custom interfaces include a dedicated microprocessor and custom firmware, and are controlled using RS232 TTY lines connected to the PDP-11/44. The attendant console interface was developed at Lincoln. It, in effect, pushes buttons and monitors the lights and alphanumeric display on the attendant console. This allows the RCP to route calls and monitor their status. The Tone TX/RX interface was developed at Wescom, Inc. under subcontract to Lincoln. It is used to send and receive the Dual-Tone-Multi-Frequency (DTMF) tones produced by a Touch-Tone keypad when an RCP number is dialed. The E&M and Central Office trunk signaling controllers are necessary to monitor the status of interswitch trunks, to preempt trunks, and to establish dial-and-hold trunks through the Bell System long-distance network. Hardware for these interfaces was developed at Wescom, and firmware is being developed jointly at Lincoln and Wescom.

### 3.3.3 Software

Software for the RCP is being written in the C language and run under a real-time version of the UNIX* operating system called VENIX. This operating system was chosen because it supports the real-time features needed for the RCP but also provides UNIX software tools and utilities that simplify software development and documentation.

Figure 13 is a block diagram of the software structure of the RCP. It includes multiple software processes that communicate through shared memory segments provided by VENIX. Software processes include handlers to drive custom interfaces and a CCS TTY link, a foundation process to bring the system up and recover from software errors, a background process to collect statistics and provide a user interface, a switch controller to control RCP-switch interactions via high-level commands, a phantom call controller to generate emulated traffic, and an executive to implement new high-level CCS protocols and control the call-setup logic. The user interface built into the background process is designed to simplify RCP control and monitoring. The RCP is controlled by selecting and altering the values of

---

* UNIX is a trademark of Bell Laboratories.

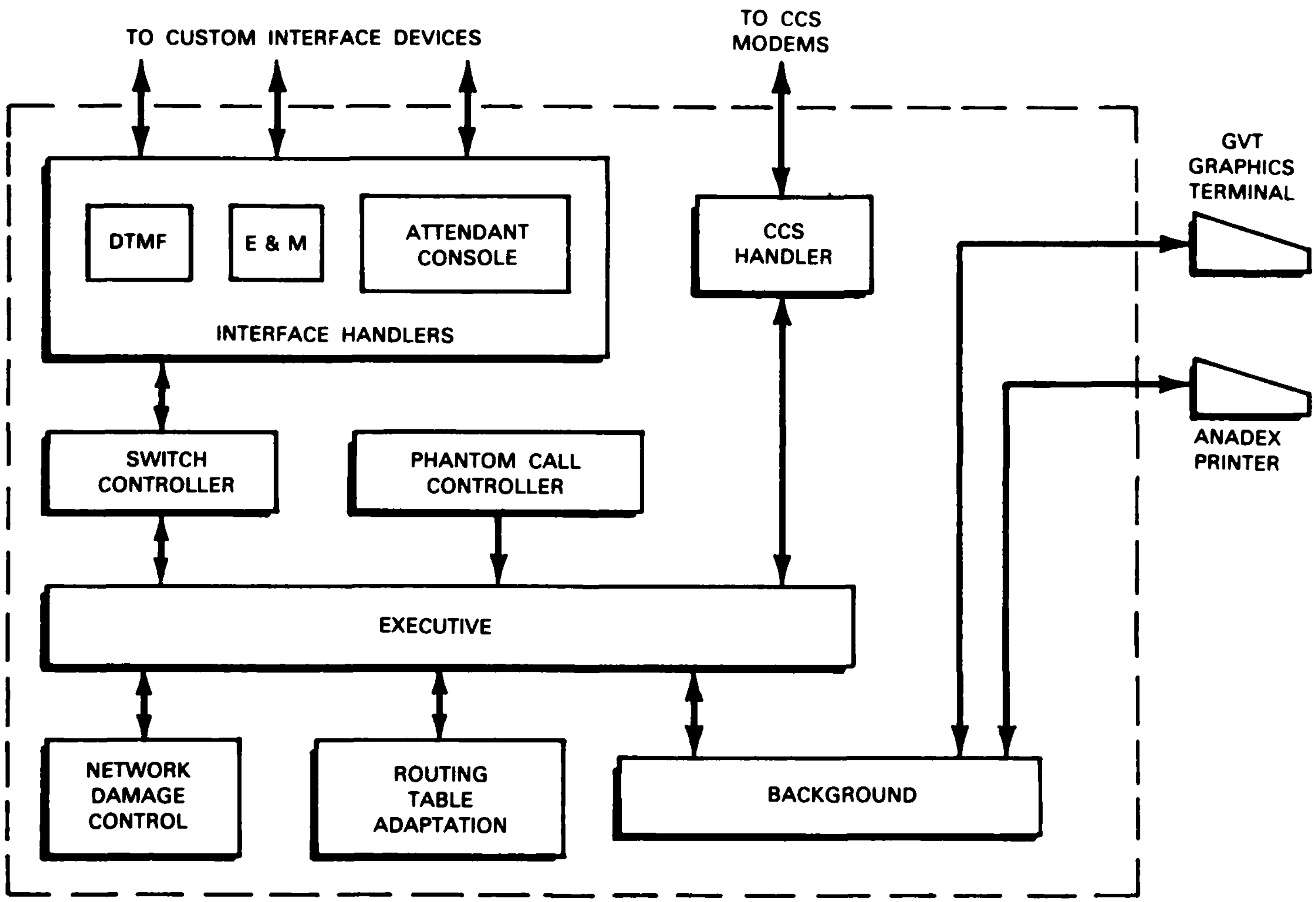



Figure 13. RCP software structure.

126558-N 01

items on menus displayed on an intelligent computer terminal. This simplifies the task of setting up simulation runs with emulated traffic, of examining statistics, of selecting routing and preemption procedures, and of altering network controls. For example, statistics and information describing the status of real and emulated traffic are displayed and updated in real time on a monitor screen. Simulation runs that use emulated traffic can be rapidly set up and initiated by changing items on a menu, and RCP behavior can be examined by watching the monitor screen as it is updated in real time.

### 3.3.4 Common-Channel Signaling (CCS)

The CCS protocol used in the RCP is implemented in both the CCS handler and in the Executive process. The CCS handler implements a link-level protocol compatible with CCITT No. 7. This handler transmits CCS messages over direct-connect RS232 TTY lines, over 1200-baud full-duplex lines obtained using modems and dialed-up Bell System phone connections, or over 1200-baud full-duplex virtual circuits obtained using PCIs and the wideband network. One such line will parallel each group of voice trunks in RCP networks. The protocol used in the CCS handler differs from the CCITT standard only in that asynchronous 1200-baud connections are used for CCS links instead of synchronous 64,000-baud connections. The Executive implements a new telephone-user-level protocol modeled after the protocol used in the call-by-call simulator that is again compatible with CCITT No. 7.

## 3.4 FUTURE RCP INSTALLATION PLANNING

Reference was made last year to switch types that will be available at the other EISN sites to act as part of the advanced EISN test bed. In particular, we noted that the choice of the DBX-1200 was based upon the fact that Ft. Huachuca had in place a new DBX-5000, carrying large amounts of operational traffic, which was a prime candidate for inclusion in EISN. Since then, we have decided that it would probably be imprudent to jeopardize operational traffic in any way by introducing EISN interfaces into the Ft. Huachuca switch (now renamed the UTX-5000). No decision has yet been made on the switch type that will actually be used with EISN at Ft. Huachuca; however Lincoln recommends that a small low-cost switch such as the UTX-1200 be purchased for the purpose. Ft. Monmouth intends to use a Northern Telecom SL1; it was found that spares and expansion units purchased by Ft. Monmouth when their new PBX (an SL1) was installed some time ago almost constitute a complete switch that could be used by EISN, and that only a modest amount of other equipment need be purchased to complete the switch. Last year we noted that the switch type to be used at RADC was unknown, but that a SCOPE DIAL switch (the Northern Telecom DMS-100) was a good possibility. At present, there is still no definite decision on this point.

During FY 84, Lincoln intends to work with Northern Telecom engineers to design an SL1 interface which is functionally equivalent to the attendant console interface on the UTX-1200. The necessary hardware and software will be prepared at Lincoln, and delivery of a RCP and custom interfaces will be made to Ft. Monmouth early in FY 85. Similar functions will be carried out by Lincoln for the Ft. Huachuca and RADC switches after decisions have been made on switch selection for those sites.

# 4. DATA PROTOCOLS AND VOICE/DATA INTEGRATION

The goal of our work in the area of data protocols is to explore the performance of the DoD standard protocols, TCP and IP. IP (the Internet Protocol) supports the delivery of datagram packets in an internet made up of heterogeneous networks interconnected with gateways. IP makes no guarantee to deliver an offered datagram or to deliver in order those that do make it through the internet. To provide a service of the sort needed for file transfers or interactive terminal communications, an additional protocol layer is needed. TCP (the Transmission Control Protocol) provides those needed services as well as end-to-end flow control to prevent the receiver of a packet stream from being overwhelmed by a flow greater than it can handle. TCP makes use of IP which, in turn, makes use of whatever local network protocols apply in the actual networks to which hosts or gateways are connected. During the current fiscal year, these protocols have become operational in the ARPANET and the other DoD-supported networks that are interconnected to form the DoD Internet. Performance information is being gained from the hosts and network nodes in this operational data-only environment. Our work is aimed at extending the knowledge of data protocol performance to a voice/data environment and at carrying out experiments in an environment where network characteristics can be controlled to explore the interactions between protocol design options and network characteristics such as flow capabilities, delays, and control policies.

During FY 83, we have developed measurement tools and extended the capabilities of our IP/ST gateways to support data protocol experiments. (ST refers to the experimental internet Stream protocol.) The measurement tools are embodied in measurement hosts (MHs) which are packet-voice terminals with the addition of timer cards that provide a globally synchronized time base for cross-net delay measurements and the control of packet dispatch intervals. Currently, there are two software packages available for the MHs. One provides for the generation and measurement of either IP datagrams with deterministic or Poisson traffic patterns or of ST protocol packets with a multi-talker talkspurt traffic model. We use this software package primarily to provide background traffic for tests with the second package that generates and measures traffic with a TCP file-transfer model. We use the TCP file-transfer model rather than an interactive terminal model for two reasons. The statistics of a terminal model are not well understood, and the interactions between the TCP control options and the network characteristics are more readily measured for the sustained file transfer task than for the sporadic terminal interactions.

Changes to the IP/ST gateways have extended the IP capabilities to include fragmentation and the generation of all Internet Control Message Protocol (ICMP) messages that are needed for our experiments. The gateways can now send IP datagrams either in WB SATNET streams or as WB SATNET datagrams under the control of the experimenter. The basic difference in the two types of WB SATNET service is an increased delay of at least one satellite round trip for the datagram service relative to the stream. This disadvantage

may be somewhat compensated by a greater flow capability for the slower service if voice traffic is using most of the stream capacity. The gateways have also been augmented to accumulate histograms of stream capacity remaining after voice packets have been dispatched and again after data packets have been dispatched.

Flow control in the gateways is determined by the WB SATNET stream parameters that specify the maximum number of packets that can be dispatched in one stream interval and the total quantity of data that can be transmitted in the interval. Congestion control is provided by limiting the time (in dispatch intervals) that voice packets can remain buffered in the gateway and by limiting the size of the data-packet queues for each network. By manipulating these control parameters, we can realize a variety of apparent network characteristics by concatenating gateway-to-gateway hops with different control settings on each hop.

Opportunities for measurements on real networks have been limited during FY 83 by the activities of the wideband network Task Force in their efforts to improve WB SATNET performance and reliability. Our initial plan had been to carry out a series of experiments comparing the characteristics of the loop paths between Lincoln and DCEC via the WB SATNET and via the ARPANET and EDN. In the early part of the year, we had no regular ARPANET connection for the IP/ST gateway at Lincoln. In the latter part of the year the ESI at DCEC has not been operable, so that we have instead carried out experiments using the two IP/ST gateways at Lincoln either directly connected or looped through the satellite channel. From these experiments we have been able to draw some preliminary conclusions that we present in the next section. However, the reader should be warned that the apparent network environments which we can simulate with this two-gateway configuration are very simple, and that experiments with richer environments may well cause us to reach different conclusions in future work.

Our work in FY 84 will focus on the development of an adaptive TCP implementation and its testing in a variety of network environments. We expect to use the file-transfer application principally, but plan some tests to measure the effects of the adaptive mechanisms on interactive terminal applications as well.

### 4.1 THE TCP FILE-TRANSFER MODEL

Figure 14 shows a schematic representation of the protocol layers involved in the TCP file-transfer process. In a real file transfer, the File-Transfer Protocol (FTP) at the highest level takes care of the details of naming conventions and file formatting appropriate to the file systems at the sending and receiving hosts, and sets up a TCP connection for the actual transmission of the file. In our experiments, we are concerned only with the network and protocol actions relating to the file transfer itself and do not simulate any FTP-level activity. Our simulations begin at the point where a connection has already been set up and the transfer itself is ready to start. Further, we assume that the file contents can be made available to the sending TCP as fast as the data can be sent, and that on the receiving end the file contents can be absorbed from the TCP layer at the rate that they arrive. In a real

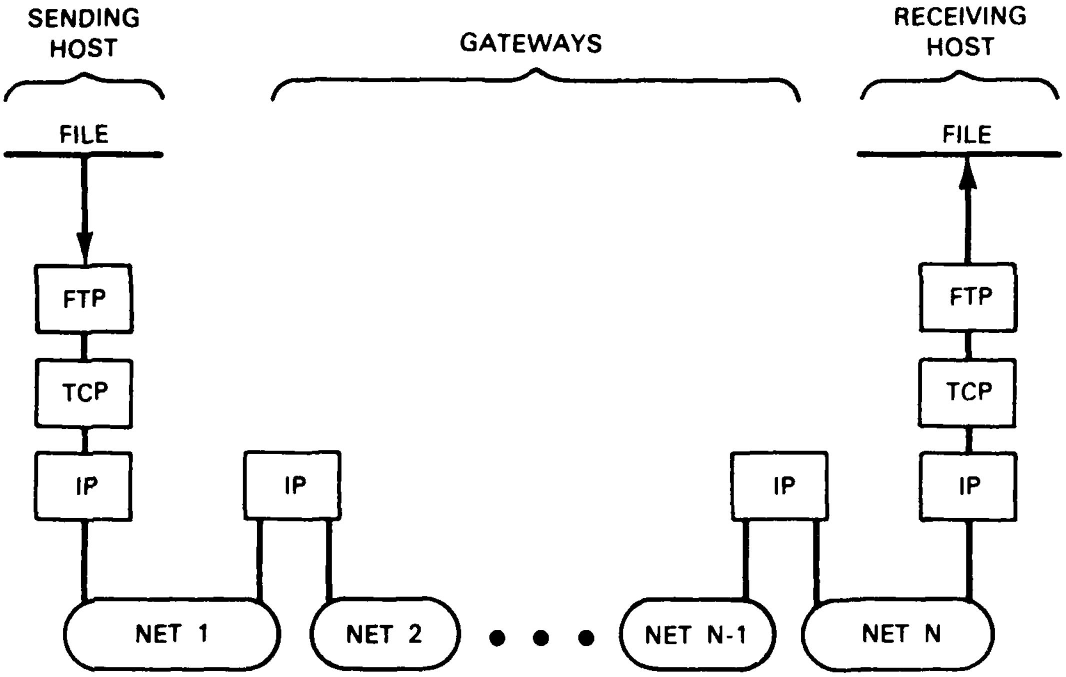



Figure 14. TCP file-transfer model.

file transfer, these assumptions would not always be met, and the actual transfer rates would be lower than those that we measure, but we have no basis for a more complex model that would try to simulate the behavior of real operating systems, and we are not interested in matching any particular real-world situation. Rather, our goal is to explore the interaction between protocol parameters and network characteristics.

The TCP layer has two functions. The first is to take a continuous stream of data from the sending FTP and deliver that stream to the receiving FTP in order and without any gaps due to lost packets or bit errors. The second function is to control the rate of flow to a value that is acceptable to the receiving FTP. To support the first function, the TCP protocol provides a numbering scheme for the data stream to order arriving packets, a packet checksum for data integrity, and an acknowledgment and retransmission mechanism to deal with lost or damaged packets. The numbering scheme is a count of octets of data bits starting from an agreed-upon starting number. Each message sent by the TCP layer (called a "segment") carries the octet number of the first octet carried in the message. The acknowledgments (ACKs) carry the octet number of the next expected octet in the stream (one more than the number of the last good octet received). If a TCP connection carries data in both directions, as would be the case for terminal communications, the ACKs may

be piggy-backed in real data segments moving in the opposite direction, but for our file-transfer model there are no data moving in the reverse direction, and ACKs must travel as ACK-only segments.

To support the end-to-end flow-control function, TCP provides a parameter called the "window" that is specified by the receiver and sent in every segment, real data or ACK-only. The window specifies the number of octets that can be transmitted beyond the ACK'ed octet number. From the receiver's point of view, the window size represents a quantity of buffer space that the receiver is prepared to commit to ordering packets which may arrive out-of-order due to network behavior. From the sender's point of view, the window size represents an upper limit on the buffering it may need in order to be able to retransmit lost or damaged segments. From the point of view of data transfer rate, the window size limits the quantity of unacknowledged octets that can be in flight between sender and receiver. If the sending TCP obeys the window convention, the maximum average transfer rate will be the window size divided by the average round-trip time (RTT). The RTT is defined as the time between the sending of a segment and the receipt of an ACK covering the segment. The limitation on transfer rate posed by window size can be severe for wideband long-delay networks such as the WB SATNET. For example, a window size of 4000 octets, which might be viewed as generous by many host computers, would limit the transfer rate to about 2500 octets per second for datagram service across the WB SATNET. This rate is roughly two orders of magnitude below the maximum that the WB SATNET could sustain, but it is not reasonable for a receiving host to offer a window size large enough to support such a high rate.

In conditions where the window rather than the capacity of the network limits the transfer rate, a behavior called "silly window syndrome" has been observed. This behavior results from an unfortunate interaction between the receiver's ACK policy and the sender's use of the available window space. When the syndrome occurs, the data end up flowing in packets that are much smaller than would be desired for the network characteristics with a consequently severely reduced rate of flow. A discussion of the syndrome and various proposed cures is given in Reference 8. Although we are interested primarily in network-limited rather than window-limited experiments, we cannot ignore the silly window case and must design our experiments so that the results will not be distorted by silly window effects.

A further use of the window parameter is to curtail flow altogether should the receiving process (FTP in our model) be unable to keep up with the rate of data arrivals. In such a case, the receiving TCP can send a window size of zero to effectively stop transmission (after a delay) by the sending TCP. We are not concerned with this case in our simulations, and always maintain a fixed window at whatever value the experimental conditions warrant.

The sending FTP has the task of picking a segment size for its transmissions. Common sense suggests that the segment size should be set as large as possible, i.e., as large as

either the window size or the maximum packet size of the networks permit. Such a policy would tend to minimize costs in networks that base their charges on number of packets sent and lead to efficient utilization of resources in networks where explicit charging for services is not done. A host can easily know the maximum packet size acceptable to the networks to which it may be directly connected, but there is no straightforward way for it to determine the maximum size that can get through an arbitrary internet path without either being dropped or fragmented by some gateway to fit into a net with a maximum size smaller than that of the segment size. Fragmentation, when it occurs, is carried out by the IP layer in a gateway along the route. Reassembly of the fragments is handled by the IP layer in the receiving host. If packets are not lost in the network, fragmentation will have no significant effect on the transfer rate. However, if losses occur beyond the point of fragmentation, more retransmissions will be needed than would be the case if the segment size was reduced to a point where fragmentation does not occur. The sending TCP can ask the IP layer to mark the segments as not fragmentable. In such a case, should the segment be too large for a network, it will be discarded and an ICMP message will (with some probability) be returned to the sending TCP informing it that the packet was dropped. The sending TCP could then retransmit the data in a smaller segment and keep reducing the size until no packet-too-large ICMPs were being received. In practice, a TCP implementation is likely to pick a maximum segment size that is small enough to make it through most nets and permit fragmentation to deal with the few cases that fail.

The sending TCP also has the task of choosing the rate at which it dispatches segments. The maximum transfer rate that can be achieved will be limited either by the window size or by the capacity of the network path. If packets are dispatched at a rate greater than the network can sustain, congestion control mechanisms in the gateways and/or nets will discard some of the packets. The IP specifies that a gateway should send an ICMP "SOURCE QUENCH" message back to the sender of an IP datagram when a packet is discarded due to overload of network or gateway resources. Obviously, there can be no guarantee that such messages will make their way back in all cases; but, should they arrive, the sending TCP will get some information about the capacity of the network path at some time in the recent past.

The IP specification recommends that the sender reduce his transmission rate in response to the quench messages to help relieve the congestion, and this behavior can be used by the sending TCP to reduce its dispatch rate until quench messages are not being received at an excessive rate. This feedback mechanism can be effective in throttling flow, but its effect on flow rates is entirely negative. A sending TCP needs to probe the state of the internet by constantly trying to increase its rate if the maximum useful transfer rate is to be obtained.

Whether the transfer rate is limited by the window size or the network capacity, common sense and our experience suggest that the segments be spaced over the RTT to avoid unnecessary congestion and consequent packet loss that is more likely to occur if the transmissions are bunched together. A contrary point of view has been argued by D. Clark.[8]

who recommends sending a window's worth of data in a burst to make better use of the resources of large time-shared host computers. Such a procedure may work when the host is connected to a net such as ARPANET that applies backpressure to limit transfer rates, but it is sure to fail for our measurement hosts connected to cable nets. In our case, bursts beyond a limited size will result in immediate packet losses and would be totally ineffective in achieving a high transfer rate.

The receiving TCP, in addition to specifying the window size, decides the acknowledgment policy. TCP ACKs are cumulative, i.e., an ACK acknowledges the receipt of all segments up to the octet pointer in the ACK message. The arrival of an out-of-order packet does not cause the value of the ACK pointer to advance. A simplistic TCP implementation might send an ACK for every arriving segment, whether or not the pointer advanced, and this policy would provide the sending TCP with maximum information about the situation at the receiver; but, as Clark points out,[8] frequent ACKing is inefficient in its use of host processing and network resources and can encourage the silly window syndrome. He recommends a policy that delays the sending of ACKs until either a timeout based on a segment arrival rate estimate by the receiver has occurred or a significant fraction of the window has been used. Comparison of the effects of these ACK policies is still to be explored in our experiments.

The sending TCP has the task of determining a retransmission policy to deal with damaged or discarded packets. The protocol requires that a timeout mechanism be used to resend segments that have not been ACK'ed in a reasonable time interval after their transmission. The proper setting for the timeout is enough longer than the mean RTT to give most of the ACKs a chance to be received before a retransmission occurs. If the timeout is too short, unnecessary retransmission will occur. If it is too long, and the window space becomes exhausted, the data transfer rate will be lower than it might have been. The sending TCP can compute an estimate of the mean RTT and its dispersion by observing the arrival time of the ACKs that cover each transmitted segment. The computation of the RTT is complicated by the fact that ACKs are cumulative (all segments up to the octet value in the ACK are ACK'ed by any ACK) and are not required to be sent for each segment received. The estimated RTT will generally have a large dispersion relative to its mean value.

Retransmission can be triggered by events other than timeouts. The arrival of a "SOURCE QUENCH" ICMP message indicates that a packet has been discarded by a gateway along the route. If the quenched segment is retransmitted at the earliest opportunity instead of waiting for it to time-out, there may be some increase in the transfer rate. Similarly, the arrival of two or more ACKs with the same octet pointer is likely to indicate that segments have been lost, and this event could be used to trigger retransmission.

In the usual TCP implementation, the sender maintains a queue of segments awaiting acknowledgments, and the queue information includes time information for triggering

retransmission on timeout. If the RTT is long in relation to the interval between transmissions, as it is in many cases of interest, then, once the timeout condition is met for a single lost segment, it will continue to be met for succeeding segments until either the window limit is reached or an ACK arrives indicating successful retransmission. Consequently, the loss of a single segment results in a burst of retransmissions of roughly one RTT duration. If the probability of packet loss in the network is low, it would appear to be better to retransmit only the first of the unacknowledged segments and resume transmission of new segments as long as window space was available.

If fragmentation occurs because the transmitted segment size is too large, reassembly takes place in the IP layer in the receiving host. If some of the fragments are lost, the retransmission mechanism is not as efficient as it is when a comparable fraction of full segments is lost. The problem is that arriving fragments can contribute to the file transfer only when all the fragments of a segment arrive. If a segment is retransmitted and fragmentation occurs again, the fragments of the retransmitted segment are not identified by the receiving IP layer as being related in any way to the fragments it may have on hand from the original transmission. Consequently, fragmentation can be very damaging when the probability of loss is significant. Unfortunately, the TCP provides no mechanism for a receiver to use to inform the sender that fragmentation is occurring. If such were available, the sender could reduce the segment size to avoid the problem.

### 4.2 AN ILLUSTRATIVE EXAMPLE EXPERIMENT

In this section, we describe one of our preliminary experiments as an example of the kind of exploration that our current capabilities can support. The experiment was intended to examine the effect of varying the pacing of TCP segment transmissions in a very simple network environment. The experiment makes use of the configuration shown in Figure 15. The network consists of a single gateway-to-gateway hop which can have a relatively short delay obtained by connecting the gateways with a cable (the dashed line in the figure) or a long delay by connecting them through the PSAT and instructing it to send the packets to the satellite. With light to moderate loads, the RTT for the short-delay situation varied from 140 to about 220 ms. Similar loads gave RTT values of 780 to 1020 ms for the satellite loop. Other delay situations between these extremes could have been obtained by sending only the data segments and not the ACKs to the satellite, but the results from the extreme cases suggested that in-between values would not be worth exploring.

Flow in the simulated network was controlled by defining a stream reservation with a capacity to carry 2 data packets every 44 ms (45.5 packets per second). The stream reservation also specified the total number of bits that could be transmitted, but the stream capacity and the sizes of test packets were chosen so that the only limitation on flow was the number of packets that could be carried.

A background traffic flow was generated by a pair of measurement hosts sending packets of a constant size but at a time-varying rate that followed Poisson statistics. The

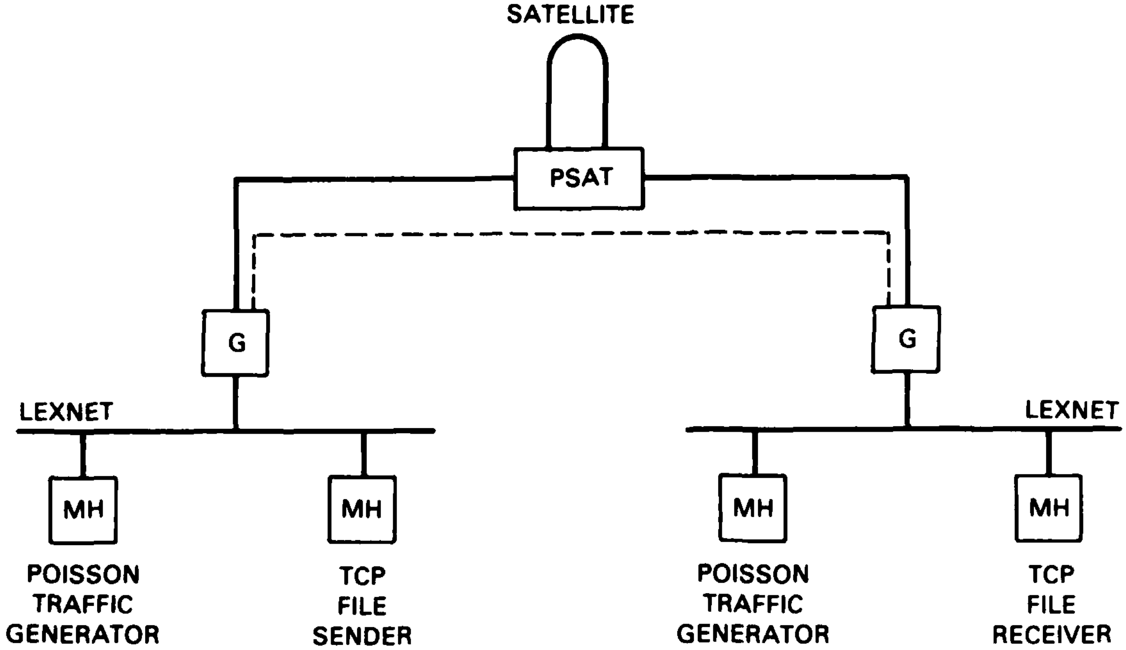



**Figure 15. TCP file-transfer experiment configuration.**

average rate of the Poisson generators was 33.3 packets per second, so that in the absence of any contention from the TCP file-transfer experiment, they would use 73 percent of the available capacity on the average.

Buffering in the gateway was under control of the experimenter and, for the results we will show here, was fixed at 10 packets. Packets arriving to find a full buffer were discarded, and an ICMP "SOURCE QUENCH" message was sent back to the source of the discarded packet. No use was made of these quench messages except to count them to determine how many packets were dropped by the simulated network. A packet arriving to find an almost-full buffer would experience a worst-case queueing delay of 220 ms. The buffer of 10 packets was sufficiently large that no losses occurred on the peaks of the Poisson flow in tests of a few-minutes duration.

For the TCP file-transfer test, the window size was chosen to be sufficiently large that the transfer was never window-limited. ACKs were generated for every received packet that advanced the acknowledgment pointer (not for duplicates resulting from wasted retransmission or for out-of-order segments). Retransmission occurred only on timeouts, and the timeout value remained constant at 5.0 s for all experiment runs. Other experiments showed that the timeout value had little effect on throughput unless it was set to be shorter than the maximum RTT or long enough to exhaust the window and cause transmission to stop until the timer triggered retransmission. The 5.0-s value was not long enough to exhaust the window.

The sending TCP dispatched segments at a constant rate set by the experimenter. When the time to dispatch a segment arrived, the program would check to see if the retransmission timeout for a previously sent segment had occurred. If it had, a retransmission would occur. Otherwise, a new segment would be dispatched as long as doing so would not exceed the window space. A consequence of this pacing mechanism is that, once retransmission had started due to a missing ACK, all dispatched segments were retransmissions until an ACK arrived that moved the acknowledgment pointer. For low packet loss rates, such an ACK was likely to arrive in a little more than one RTT after the first retransmission. Therefore, a burst of retransmissions of about one RTT duration resulted from each lost packet.

The combination of ACKing every arriving segment at the receiver and sending at a rate substantially faster than one segment every RTT causes the file transfer to be very resistant to the loss of ACKs on the reverse path. Because the TCP ACK is cumulative, the ACK to a succeeding segment satisfies the function of a lost ACK, and the transfer proceeds without detectable change in speed for moderate rates of ACK loss. This observation is in agreement with the recommendation by Clark[8] that ACKs be sent less often than one per received segment to improve network and operating system efficiency and to help avoid the silly window syndrome. He suggests sending ACKs in the file-transfer case only when a substantial fraction of the window has been used or when the receiver detects an apparent gap in the flow of segments. Clearly, if fewer ACKs are sent, the impact on the flow of lost ACKs would be greater. Additional experimentation is suggested to examine the interactions between ACK losses and Clark's suggested ACK policy.

In our experiment, file segments always contained 200 octets of data in addition to the headers. This size was picked as a convenient one for the present measurement host implementation. A larger segment size would have resulted in a proportionately larger transfer rate so long as the size did not become large enough to cause fragmentation. The window was set to a multiple of the segment size so that no odd-sized segments would be sent to use up the last little bit of window space.

Figure 16 shows a plot of the file-transfer rate as a function of the rate at which segments were dispatched by the sending TCP. Here and in Figures 17 and 18, the solid curves show results for the case of the satellite loop. The dashed curves apply to the direct (low-delay) connection between the gateways. Figure 17 shows the rate of packet loss in the sender-to-receiver direction as a function of the dispatch rate. The stream parameters were adjusted so that no losses occurred in the reverse direction. Figure 18 shows the packet efficiency of the forward path, which we define to be the fraction of packets delivered to the receiver that contributed to the file-transfer process. Retransmissions that were duplicates of segments already received use network capacity and do not contribute to the transfer. As the dispatch rate increases, such retransmissions occupy more and more of the available transmission opportunities and efficiency drops.

Taken together, Figures 16 and 17 indicate that the maximum rate of file transfer occurs in a region where a few percent of the offered traffic is being dropped by the

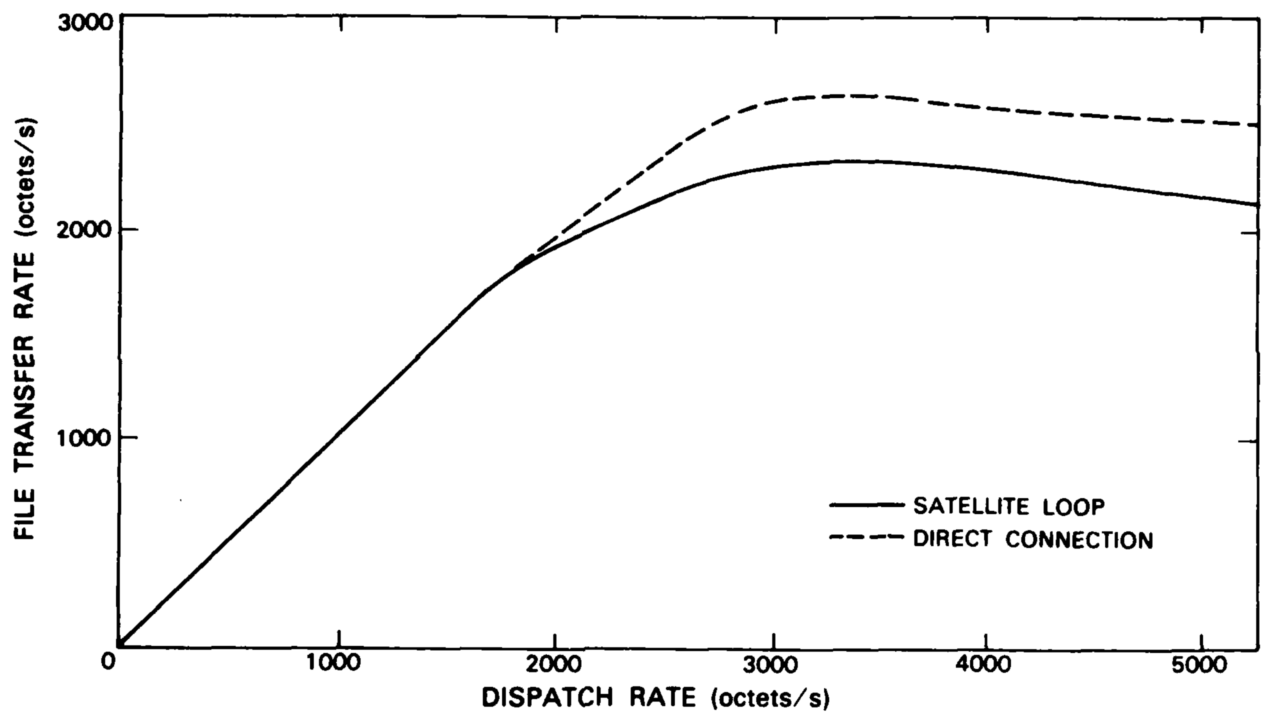



**Figure 16. Plot of file-transfer rate vs dispatch rate for TCP file-transfer experiment.**

135499-N

135500-N

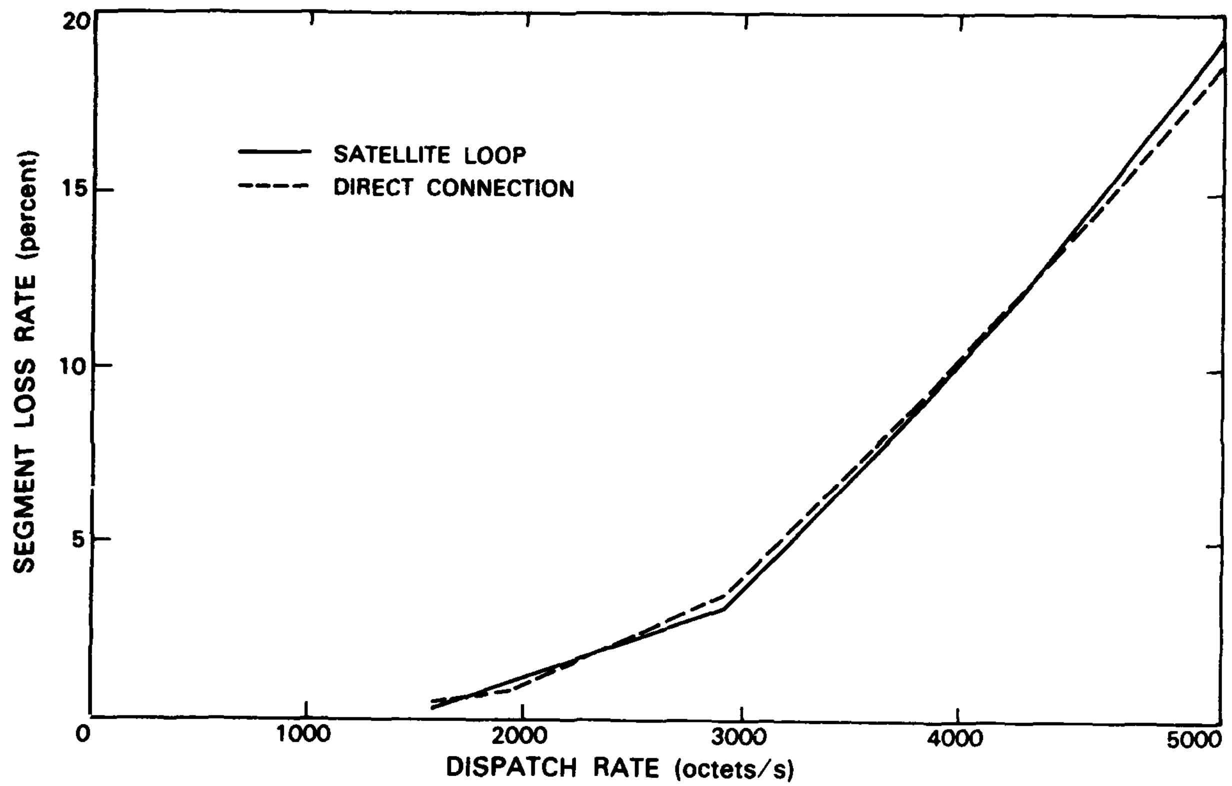



Figure 17. Plot of segment loss rate vs dispatch rate for TCP file-transfer experiment.

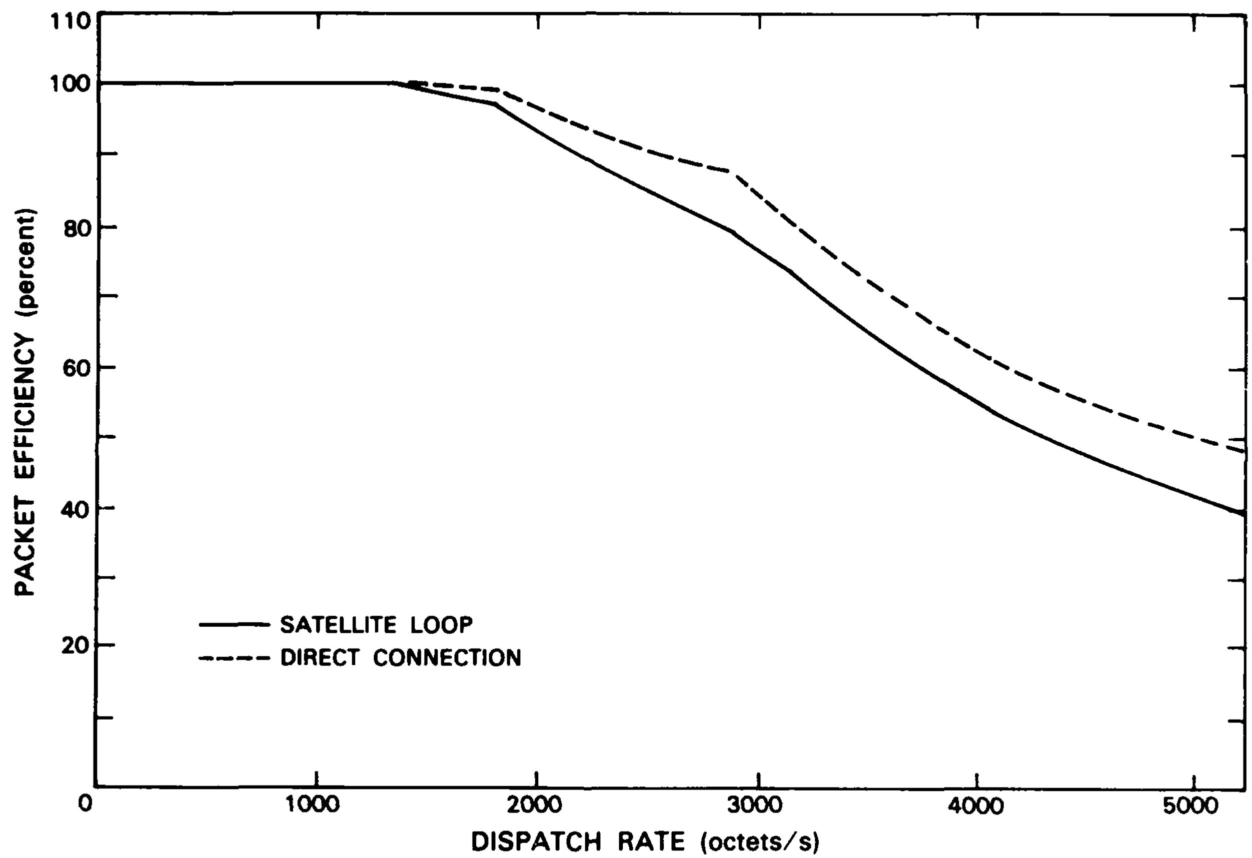



**Figure 18.** Plot of packet efficiency (ratio of useful packets delivered to total packets delivered) vs dispatch rate for TCP file-transfer experiment.

135501-N

network. For the short-delay network, this rate is roughly twice the highest rate that could be achieved without any lost segments. The difference in the rates between the long- and short-delay cases is due to the longer burst times for retransmissions in the satellite loop case. The results suggest that if the sending TCP were to use a retransmission policy that caused only the first packet in the burst to be retransmitted when a timeout was triggered and allowed new transmissions to succeed that one retransmission, the differences for the delay cases would disappear and packet efficiency would improve significantly. Clearly, such a strategy would be effective only when losses were infrequent and likely to be isolated.

It should be noted that, with the model we are using, retransmissions do not increase the offered traffic. The retransmissions merely substitute for the transmission of new segments. In the experiments, when losses are occurring for the TCP file transfer, the Poisson traffic is experiencing a similar rate of loss and it, too, does not increase its offered rate as a result of the losses.

A conclusion we draw from this experiment is that an adaptive TCP implementation seeking to maximize the file-transfer rate should adjust its dispatch rate to a value such that a few percent of the offered segments are being discarded by the internet gateways. Of course, if the network to which the sending host was connected exerts backpressure, as does the ARPANET for example, it may not be possible for the TCP to reach a dispatch rate at which segments are being dropped by gateways beyond its local net. In our experiment, the LEXNET to which the measurement host was connected did not exert backpressure at the rates we used.

An internet in which users were aggressively following this adaptive policy would tend to operate with its communication channels fully loaded. A small percentage of packets would be lost and some capacity would be wasted with unnecessary retransmissions. If the number of wasted retransmissions can be kept low, the overall utilization of resources could be better than would be achieved if the average offered traffic were reduced to a point where no losses were occurring. A point to be noted is that, when the average traffic approaches or exceeds the channel capacity, very little buffering is needed to keep the channel fully loaded, and that buffering beyond that point merely adds to overall delay.

### 4.3 VOICE/DATA INTEGRATION

If we substitute a voice background traffic for the Poisson data traffic in the experiment described above, we get a similar result. The maximum transfer rate occurs at a point where some packets are being dropped. However, the channel cannot be fully utilized unless the voice and data segments happen to be the same size and the stream capacity is adjusted for an exact fit to some multiple of this size. For effective use of the channel capacity with a more general voice/data mix, it is necessary to fragment the data segments to fit the leftover stream slots. Good performance for voice traffic requires that packets be dispatched to a channel at frequent intervals and, consequently, in small chunks. As a result, the leftover

channel available for data traffic has many opportunities for sending small packets but very few opportunities for sending large ones. These opportunities may be well suited to carrying interactive data traffic but certainly are poorly matched to a file-transfer load that, to be efficient, needs large packet sizes. Fragmentation of the large data packets to fit the available transmission opportunities works well for utilizing the spare capacity, but it causes an unfortunate increase in the number of packets that have to be handled by nodes downstream from the point of fragmentation. This problem can be avoided by reassembling the fragments in the gateway at the receiving end of the channel for which the fragmentation is occurring.

We have implemented IP fragmentation to fit stream capacity in our IP/ST gateways, and histograms of leftover capacity accumulated in the gateways have shown that the capacity can be effectively used. Experiments at the TCP level cannot take place until either the TCP measurement host or the gateway has been extended to handle the required fragment reassembly. Both of those extensions are planned for FY 84.

# 5. EISN SYSTEM COORDINATION AND EXPERIMENT PLANNING

In April 1983, Lincoln submitted the "Defense Switched Network Technology and Experiments Work Plan, FY83" to DCEC. This Plan focused on three major experimental areas: (1) completion and experimental exploitation of a 5-node network of Experimental Integrated Switched Network (EISN) sites with an interim routing and system control test-bed capability; (2) development and experimental application of the call-by-call network simulator; and (3) development and test of an advanced EISN routing and system control facility at Lincoln, consisting of an off-the-shelf computer-controlled switch integrated with an outboard Routing/Control Processor (RCP). The latter element included implementation at Lincoln of a second switch/RCP complex for subsequent installation at DCEC, and also included planning of RCP interfaces for Fts. Monmouth and Huachuca. In addition, the program included data internetting experiments on a packet-switched internetwork involving DCEC, Lincoln, the EISN satellite network, and the ARPANET.

The tasks set forth in the Work Plan have been carried out during FY 83, as described elsewhere in this report. Work has been in progress on extension of detailed experiment planning to FY 84, and an FY 84 Work Plan is in preparation.

Lincoln has undertaken an expanded role in wideband network system coordination during FY 83. A Wideband SATNET Task Force was established at the March 1983 Wideband Meeting, with the objectives of resolving current system problems and achieving reliable operation first at 1.544 Mbps and then at 3.088 Mbps. The Task Force is coordinated by Lincoln, and includes representatives from BBN, ISI, and LINKABIT. Some of the major results and observations with respect to the Task Force activity are:

(1) Two sites (Lincoln and ISI) are now operating at 3.088 Mbps with different coding rates for control and speech packets.

(2) The most significant progress has been made during site visits where the Task Force members convened at one site for intensive measurement and debugging efforts.

(3) The problems corrected by the Task Force were not concentrated in a single WB SATNET subsystem, and often involved detailed interactions between subsystems.

(3) The Task Force has extended its efforts to deal with integration of the new WB SATNET sites at RADC, Ft. Monmouth, and Ft. Huachuca.

When the Task Force was initiated in March 1983, an initial action was to set up coordinated site visits for intensive system-level debugging. In pursuing the Task Force activity over the ensuing weeks, it was found that the most significant progress was in fact made during the organized site visits, when the Task Force members convened at one of

the sites and devoted their efforts exclusively to measurement and debugging for several days. A total of four site visits have taken place to date, and at least one more is planned. The network status at this time is that two sites (Lincoln and ISI) are fully outfitted with the upgrades and fixes developed during the Task Force work, and are operating with reasonable stability at 3.088 Mbps with mixed coding rates (headers and control at rate 1/2, voice coded at rate 3/4). Work is in progress to upgrade the remaining sites, and to address additional system issues associated with multisite operation involving the seven WB SATNET sites.

The Wideband Network Task Force delivered a status and progress report to DARPA and the DCA on 1 August in Washington. This report detailed the objectives and accomplishments of the Task Force, focusing particularly on the three site visits that had been carried out at that time. It was recommended that the Task Force effort be continued. It was also recommended that an order for four additional ESI-As be added to the existing order for six units, so that all ten WB SATNET sites expected to be active by next year would be outfitted with them. Additional discussions at the Task Force meeting focused on longer-term issues related to controlling, verifying, and measuring subsystem availability for experimental use. These issues were network and subsystem configuration control, standardized network test procedures, methods for measuring and reporting network quality factors, and host-level testing procedures. The Task Force is currently addressing these longer-term issues as well as the specific efforts involved in system debugging and integration.

On 21-22 September, acceptance testing of the new Western Union earth stations at Ft. Monmouth and Ft. Huachuca was used as a focus for implementing new power and frequency calibration procedures that will become standard in the WB SATNET. The basic issue is that the automatic gain and frequency control (AGC and AFC) mechanisms in the ESI have finite limits on their acquisition apertures, and TDMA communications among multiple sites will fail if the site-to-site differences in these parameters exceed the limits. If local Western Union personnel use locally owned power meters and frequency counters (with possible differences in calibration) to set the parameters, it is very likely that the limits will be exceeded. Accordingly, it has been proposed that all stations be adjusted relative to a reference station, and that this reference station should be Lincoln Laboratory. A highly accurate HP8566A spectrum analyzer has been purchased and installed at Lincoln to support this function. In calibration of the Army sites, Lincoln's transmitted amplitude and frequency were first carefully measured in satellite loopback, and then each station in turn was adjusted so that its parameters (as measured with the spectrum analyzer) matched Lincoln's.

# REFERENCES


1. Network Speech Systems Technology Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1981), DTIC AD-A112718/2.
2. Defense Switched Network Technology and Experiments Program Annual Report, Lincoln Laboratory, M.I.T. (30 September 1982), DTIC AD-A128932/1.
3. R.P. Lippmann, "Steady State Performance of Survivable Routing Procedures for Circuit-Switched Mixed-Media Networks," Technical Report 633, Lincoln Laboratory, M.I.T. (29 December 1982), DTIC AD-A126213/8.
4. W.C. Miller, T.D. Walsh, J. Van Zuylen, A. Covo, and T.G. Huemiller, "Multi-Level Precedence and Preemption Study," Defense Communications Engineering Center Engineering Publication No. A04-81 (November 1981).
5. R.P. Lippmann, "Survivable Routing Procedures for Circuit-Switched Satellite-Terrestrial Networks," ICC '83 Conference Record, Volume 1, Boston, 19-22 June 1983, pp. 17-22.
6. C.J. Weinstein, "The Experimental Integrated Switched Network — A System-Level Network Test Facility," MILCOM '83 Conference Proceedings, Vol. 2, Arlington, Virginia, 31 October — 2 November 1983, pp. 449-456.
7. H.M. Heggestad and C.J. Weinstein, "Voice and Data Communication Experiments on a Wideband Satellite/Terrestrial Internetwork System," ICC '83 Conference Record, Volume 1, Boston, 19-22 June 1983, pp. 1-8.
8. D.D. Clark, "Window and Acknowledgement Strategy in TCP," Network Working Group Request-for-Comments RFC 813, July 1982.

# GLOSSARY

| | |
|---|---|
| ACK | Acknowledgment |
| CCS | Common-Channel Signaling |
| CPU | Central Processing Unit |
| DAMA | Demand-Assignment Multiple Access |
| DCA | Defense Communications Agency |
| DCEC | Defense Communications Engineering Center |
| DSN | Defense Switched Network |
| EISN | Experimental Integrated Switched Network |
| ESI | Earth Station Interface |
| FR | Forward Routing |
| FTP | File-Transfer Protocol |
| ICMP | Internet Control Message Protocol |
| IP | Internet Protocol |
| IPG | Internet Packet Gateway |
| MFR | Modified Forward Routing |
| MH | Measurement Host |
| MLP | Multi-Level Precedence |
| NVP | Network Voice Protocol |
| PCI | Packet/Circuit Interface |
| PSAT | Pluribus Satellite Interface Message Processor |
| RADC | Rome Air Development Center |
| RCP | Routing/Control Processor |
| RTT | Round-Trip Time |
| SATNET | Satellite Network |
| ST | Stream Protocol |
| TCP | Transmission Control Protocol |
| TOE | Telephone Office Emulator |
| WB SATNET | Wideband Satellite Network |

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>ESD-TR-83-228 | 2. GOVT ACCESSION NO.<br>AD-A140 609 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>Defense Switched Network Technology and Experiments Program | | 5. TYPE OF REPORT & PERIOD COVERED<br>Annual Report<br>1 October 1982 — 30 September 1983 |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>Clifford J. Weinstein | | 8. CONTRACT OR GRANT NUMBER(s)<br>F19628-80-C-0002 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Lincoln Laboratory, M.I.T.<br>P.O. Box 73<br>Lexington, MA 02173-0073 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>Program Element No. 33126K |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Defense Communications Agency<br>8th Street & So. Courthouse Road<br>Arlington, VA 22204 | | 12. REPORT DATE<br>30 September 1983 |
| | | 13. NUMBER OF PAGES<br>60 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)<br>Electronic Systems Division<br>Hanscom AFB, MA 01731 | | 15. SECURITY CLASS. (of this report)<br>Unclassified |
| | | 15a. DECLASSIFICATION DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

None

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

| | | |
|---|---|---|
| routing algorithms | Defense Switched Network | circuit-switched network |
| Experimental Integrated Switched Network | mixed-media routing | packet switching |
| | pre-emption | |

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report documents work performed during FY 1983 on the DCA-sponsored Defense Switched Network Technology and Experiments Program. The areas of work reported are: (1) development of routing algorithms for application in the Defense Switched Network (DSN); (2) instrumentation and integration of the Experimental Integrated Switched Network (EISN) test facility; (3) development and test of data communication techniques using DoD-standard data protocols in an integrated voice/data network; and (4) EISN system coordination and experiment planning.

8